robotron

Anleitung für den
Bediener

# Dienstprogramme

Arbeitsplatzcomputer A 7100 Betriebssystem SCP 1700

Anleitung für den Bediener

# Dienstprogramme

AC A7100


**VEB Robotron-Projekt Dresden**

C 1015-0001-1 M 3030

Die vorliegende Systemunterlagendokumentation, Anleitung für den Bediener Dienstprogramme , entspricht dem Stand von 6/86.

Nachdruck, jegliche Vervielfältigung oder Auszüge daraus sind unzulässig.

Die Ausarbeitung erfolgte durch ein Kollektiv des VEB Robotron-Elektronik Dresden .

Herausgeber:

VEB Robotron-Projekt Dresden 8010 Dresden, Leningrader Str. 9

(C) 1986

Kurzreferat

In der vorliegenden Anleitung für den Bediener des SCP 1700 werden die zum Grundpaket des Betriebssystems gehörenden Systemprogramme beschrieben. Die einzelnen Programme werden bezüglich ihrer Kommandos und Fehlerreaktionen dargestellt. Die Anleitung für den Bediener ist zur Vorbereitung und Durchführung der Arbeit am Rechner bestimmt. Die beschriebenen Programme umfassen die Arbeit mit Dateien, das Aufbereiten von Programmen, das Übersetzen von Programmen, die in Assemblersprache geschrieben sind, das Bilden von Kommandodateien und das Testen der Programme.

Druck und buchbinderische Verarbeitung:
TASTOMAT, Eggersdorf · Ag 706/206/87 — 4117

Inhaltsverzeichnis Seite


| | | |
|---|---|---|
| 1. | Einführung | 7 |
| 1.1. | Start des SCP 1700 | 7 |
| 1.2. | Kommandoformat | 8 |
| 1.3. | Steuerzeichen zur Aufbereitung von SCP-Zeilen | 9 |
| 1.4. | Dateisicherung | 11 |
| 1.5. | Kopieren der SCP-Platte | 11 |
| 2. | Dateien, Platten, Laufwerke und Geräte | 13 |
| 2.1. | Dateien | 13 |
| 2.2. | Erstellen von Dateien | 13 |
| 2.3. | Benennung von Dateien | 14 |
| 2.4. | Zugriff zu Dateien | 15 |
| 2.5. | Zugriff zu mehreren Dateien | 16 |
| 2.6. | Organisation und Schutz von Dateien | 17 |
| 2.7. | Dateispeicherung auf der Platte | 18 |
| 2.8. | Plattenwechsel | 18 |
| 2.9. | Wechsel des Standardlaufwerkes | 19 |
| 2.10. | Zugriffsstatus für Laufwerke | 19 |
| 2.11. | Weitere SCP-Geräte | 19 |
| 3. | SCP-Kommandokonzept | 21 |
| 3.1. | Kommandotypen | 21 |
| 3.2. | Residente Kommandos | 21 |
| 3.3. | Transiente Dienstprogramme | 22 |
| 3.4. | SCP-Kommandoauswertung | 23 |
| 3.5. | Steuerzeichenkommandos | 24 |
| 4. | Kommandoübersicht | 25 |
| 4.1. | Formale Bedingungen | 25 |
| 4.2. | Beschreibungsform der Kommandos | 26 |
| 4.3. | Kommando ASM86 (Assembler) | 29 |
| 4.4. | COPYDISK-Kommando (Platten kopieren) | 30 |
| 4.5. | Kommando DDT86 (Dynamische Testhilfe) | 32 |
| 4.6. | Kommando DIR (Verzeichnisausgabe) | 34 |
| 4.7. | DISKSET-Kommando | 36 |
| 4.8. | Kommando ED (Editor) | 37 |
| 4.9. | Kommando ERA (Löschen) | 41 |
| 4.10. | Kommando GENCMD (CMD-Generierung) | 42 |
| 4.11. | Kommando HELP (Bedienerhilfe) | 43 |
| 4.12. | INIT-Kommando (Formatieren von Disketten) | 44 |
| 4.13. | Kommando PIP (Dateikopie) | 47 |
| 4.13.1. | Kopie einer Datei | 47 |
| 4.13.2. | Kopieren mehrerer Dateien | 49 |
| 4.13.3. | Fügen von Dateien | 50 |
| 4.13.4. | Kopieren von Dateien zu und von Hilfsgeräten | 51 |
| 4.13.5. | Modus für Mehrfachkommandos | 52 |
| 4.13.6. | Wahlweise Parameter des PIP | 53 |
| 4.14. | Kommando REN (Umbenennen) | 57 |
| 4.15. | Kommando STAT (Statusausgabe und -änderung) | 58 |
| 4.15.1. | Setzen eines Laufwerkes auf Read-Only-Status | 59 |
| 4.15.2. | Freier Speicherraum auf der Platte | 59 |
| 4.15.3. | Dateibelegung und Zugriffsmodus | 60 |
| 4.15.4. | Setzen der Dateizugriffsmodi (Attribute) | 62 |
| 4.15.5. | Anzeige des Plattenstatus | 63 |
| 4.15.6. | Anzeige der Nutzernummer der Bereiche mit aktiven Dateien | 63 |
| 4.15.7. | Anzeige der STAT-Kommandos und Gerätenamen | 64 |

| | | |
|---|---|---|
| 4.15.8. | Anzeige und Setzen physischer und logischer Gerätezuordnungen | 64 |
| 4.16. | Kommando SUBMIT (Stapelverarbeitung) | 65 |
| 4.17. | Kommando TYPE (Anzeige von Textdateien) | 67 |
| 4.18. | Kommando USER (Nutzernummer) | 68 |
| 4.19. | LDCOPY-Kommando (SCP-1700-Systemspuren kopieren) | 69 |
| 5. | SCP-Editor ED | 71 |
| 5.1. | Vorbemerkungen zum ED | 71 |
| 5.2. | Start des ED | 71 |
| 5.3. | ED-Operationen | 72 |
| 5.3.1. | Einlesen von Text in den Puffer | 73 |
| 5.3.2. | Abschließen des Editiervorganges | 74 |
| 5.4. | Basiskommandos zum Editieren | 75 |
| 5.4.1. | Positionieren des Zeichenzeigers CP | 76 |
| 5.4.2. | Anzeige der Speicherpufferinhalte | 78 |
| 5.4.3. | Löschen von Zeichen | 78 |
| 5.4.4. | Einfügen von Zeichen in den Speicherpuffer | 79 |
| 5.4.5. | Ersetzen von Zeichen | 81 |
| 5.5. | ED-Kombinationskommandos | 82 |
| 5.5.1. | Positionieren des Zeichenzeigers | 82 |
| 5.5.2. | Textanzeige | 83 |
| 5.5.3. | Editieren | 83 |
| 5.6. | Fortgeschrittene ED-Kommandos | 84 |
| 5.6.1. | Positionieren des Zeichenzeigers und Textanzeige | 84 |
| 5.6.2. | Suchen und Ersetzen von Zeichenketten | 85 |
| 5.6.3. | Transport von Textblöcken | 88 |
| 5.6.4. | Dateisicherung und ED-Ende | 89 |
| 5.7. | ED-Fehlermitteilungen | 91 |
| Anlage 1 | KOI7- und Hexadezimalkonvertierung | 93 |
| Anlage 2 | SCP-1700-Dateitypen | 98 |
| Anlage 3 | SCP-1700-Steuerzeichen | 99 |
| Anlage 4 | SCP-1700-Fehlermitteilungen | 100 |
| Anlage 5 | Wortverzeichnis | 109 |
| Anlage 6 | Erweiterung der HELP.HLP-Datei | 119 |
| Sachwortverzeichnis | | 121 |

Tabellenverzeichnis


| | | Seite |
|---|---|---|
| Tabelle 1: | Steuerzeichen | 10 |
| Tabelle 2: | SCP-Dateitypen | 15 |
| Tabelle 3: | Logische Geräte des SCP | 20 |
| Tabelle 4: | Residente Kommandos | 21 |
| Tabelle 5: | SCP-Dienstprogramme | 22 |
| Tabelle 6: | Steuerzeichenkommandos | 24 |
| Tabelle 7: | DDT86-Kommandos | 33 |
| Tabelle 8: | ED-Kommandos | 38 |
| Tabelle 9: | Wahlweise Parameter des PIP | 54 |
| Tabelle 10: | Textübertragungskommandos | 73 |
| Tabelle 11: | Basiskommandos zum Editieren | 76 |
| Tabelle 12: | SCP-Steuerzeichen zur Zeileneditierung | 80 |
| Tabelle 12: | ED-Fehlersymbole | 91 |
| Tabelle 13: | ED-Mitteilungen zu Plattendateifehlern | 92 |
| Tabelle 14: | KOI7-Symbole | 93 |
| Tabelle 14: | KOI7-Symbole | 94 |
| Tabelle 15: | KOI7-Konvertierungstabelle | 95 |
| Tabelle 16: | Dateitypen | 98 |
| Tabelle 17: | SCP-1700-Steuerzeichen | 99 |

# 1. Einführung

In diesem Abschnitt werden Grundlagen des SCP 1700 behandelt, der SCP-Start, Anfangsausschriften, die Eingabe von SCP-Kommandos und die Herstellung von Sicherungskopien der SCP-Originalplatte.

Die auf Magnetplatten (floppy-disk, hard-disks und ähnlichen Medien) gespeicherten Informationen werden bei der Arbeit mit SCP in Programm- und Datendateien unterteilt. Die Dateien können von einer Platte in den Arbeitsspeicher des Rechners oder zu einem peripheren Gerät, z.B. zu einem Drucker, übertragen werden. Solche Aufgaben werden durch Kommandoeingaben des Bedieners, die zur Abarbeitung entsprechender Programmteile führen, gelöst. Die im Arbeitsspeicher befindlichen Programme werden nach ihrem Start als eine Folge von Befehlen abgearbeitet. Mit SCP können sowohl eigene Programme erzeugt als auch anderweitig zur Verfügung gestellte Programme genutzt werden.

## 1.1. Start des SCP 1700

Starten oder Laden des SCP bedeutet, SCP von der Systemplatte in den Arbeitsspeicher des Rechners einzulesen und ihm die Steuerung zu übertragen.
Nach Netzeinschalten wird die A7100-Hardeware durch das im ROM befindliche Monitorprogramm auf ihre Funktionsfähigkeit geprüft. Anschließend wird das Betriebssystem SCP 1700 von der zuvor in Laufwerk A oder B eingelegten Systemplatte geladen und gestartet. Enthält keines der Laufwerke eine SCP-1700-Systemplatte, meldet sich das Kommandoprogramm des Monitors und erwartet eine Bedienereingabe. Die Monitorkommandos sind in der Betriebsdokumentation für den A7100 beschrieben.
Durch das Monitorkommando B (Boot) kann der Bediener nach Einlegen einer Systemplatte in Laufwerk A oder B den Start von SCP 1700 veranlassen.
Nach dem Laden des SCP meldet es sich auf dem Bildschirm mit der Ausschrift:

```
SCP 1700            Version 2.2

Segment Address = nnn

Last Offset     = nnnn

System Generated (Datum)

MEMORY SIZE : (Speichergröße)
```

Bei Speichergrößen $\geq$ 512K Byte wird gefragt, ob eine Speicherplatte angelegt werden soll. Die Speicherplatte ist ein Bereich im Hauptspeicher, der logisch wie ein Externspeicher verwaltet wird. Die Nutzung der Speicherplatte als Arbeitsplatte bietet den Vorteil geringer Zugriffszeiten. Es ist jedoch zu beachten, daß die Speicherplatte durch System-Neustart zerstört wird.
Die Speicherplatte wird angelegt, wenn der Bediener die Frage

```
DO YOU WANT A MEMORY DISK E : ANALOG K5600.20
                              (306K BYTES FOR USER) ? (Y/<CR>):
```

mit Y (yes) beantwortet.

Nach dieser Ausschrift folgen zwei weitere Zeichen:

```
A>
```

Die Zeichen A> stellen die Eingabeanforderung des SCP dar, die Meldung des Systems. Die Meldung teilt mit, daß SCP bereit ist, von der Tastatur ein Kommando zu empfangen und daß A das aktuelle Laufwerk ist. Das bedeutet, daß SCP Programm- und Datendateien auf der Platte im Laufwerk A sucht, solange keine anderslautenden Kommandos eingegeben werden.

## 1.2. Kommandoformat

In Abhängigkeit vom jeweils über die Tastatur eingegebenen Kommando führt SCP spezifische Funktionen aus. Ein SCP-Kommando besteht aus einem Kommandoschlüsselwort , einer wahlweisen Parameterfolge und dem Endezeichen Wagenrücklauf (CR oder RETURN). Das Kommandoschlüsselwort identifiziert das Kommando und damit das durch den Rechner auszuführende Programm. Die Parameterfolge kann zusätzliche Informationen für das Kommando enthalten, z.B. den Dateinamen. Jedes SCP-Kommando ist durch Drücken der RETURN-Taste zu beenden.
Über die Tastatur eingegebene druckbare Zeichen werden auf dem Bildschirm sichtbar, dabei bewegt sich der Kursor (Positionszeiger) nach rechts. Eingegebene falsche Zeichen können durch Benutzung der Rücksetztaste oder CTRL/H wieder gelöscht werden, wobei sich der Kursor nach links bewegt.
Bei der Eingabe des Kommandowortes und der zugehörigen Parameterfolge können sowohl Groß- als auch Kleinbuchstaben verwendet werden. Im SCP werden alle Buchstaben im Kommando wie Großbuchstaben behandelt.
Die Kommandozeile beginnt im allgemeinen direkt nach der Kommandoanforderung (Systemmeldung), im SCP sind aber zwischen der Meldung und dem Kommandoschlüsselwort auch Leerzeichen erlaubt.
Im SCP gibt es zwei verschiedene Kommandoarten:

hauptspeicherresidente (built-in) und
transiente Kommandos

Residente Kommandos aktivieren ständig im Hauptspeicher verfügbare Programme, die zum SCP gehören und sofort ausgeführt werden können. Die zu Transientkommandos zugeordneten Dienstprogramme sind in Programmdateien auf der Platte abgespeichert und müssen zur Abarbeitung in den Hauptspeicher geladen werden. Im Plateninhaltsverzeichnis können die den Transientkommandos entsprechenden Programmdateien daran erkannt werden, daß hinter dem Dateinamen CMD steht. Bei Transientkommandos prüft SCP nur das Kommandoschlüsselwort. Falls ihm Parameter folgen, werden diese vom SCP ohne Prüfung an das jeweilige Dienstprogramm übergeben.

Nachfolgend wird die Bearbeitung eines residenten Kommandos durch SCP beschrieben. Das Kommando DIR veranlasst das SCP, die Namen der Plattendateien auf dem Bildschirm anzuzeigen. Nach der Systemmeldung wird das Kommando DIR ohne Parameter eingegeben und die Taste RETURN gedrückt:

```
A>DIR
```

SCP antwortet darauf mit dem Ausschreiben aller Dateinamen des Plateninhaltsverzeichnisses der im Laufwerk A enthaltenen

Platte.
Ist diese Platte z.B. die SCP-Systemplatte, so werden neben anderen Dateinamen auch die folgenden auf dem Bildschirm ausgeschrieben:

```
COPYDISK CMD
PIP      CMD
STAT     CMD
```

SCP akzeptiert nur korrekte Kommandoschlüsselworte.
Fehlerhafte Schlüsselworte werden mit einem Fragezeichen wiederholt.

Beispiel:

```
DJR statt DIR:
A>DJR
DJR?
```

Das Kommandoschlüsselwort DJR wurde nicht gefunden.

Das Kommando DIR akzeptiert als Parameter einen Dateinamen. Damit kann überprüft werden, ob sich eine spezielle Datei auf der Platte befindet.

Beispiel:

```
COPYDISK.CMD:

A>DIR COPYDISK.CMD
```

SCP antwortet entweder mit der Anzeige das Namens der gesuchten Datei oder mit der Mitteilung NO FILE.
Zu beachten ist dabei, daß ein Leerzeichen als Trennzeichen zwischen dem Schlüsselwort und dem Parameter steht.
Wird dieses vergessen, so erscheint folgende Ausschrift:

```
A>DIRCOPYDISK.CMD
DIRCOPYDISK.CMD?
```

## 1.3. Steuerzeichen zur Aufbereitung von SCP-Zeilen

Falsch eingegebene Zeichen können mit der Rücksetztaste (Backspace) korrigiert werden. Für eine effektive Arbeit werden durch SCP folgende Steuerzeichen verwendet. Diese Steuerzeichen können zur Aufbereitung von Kommandozeilen und Eingabezeilen der meisten Programme benutzt werden. Die Eingabe eines Steuerzeichens erfolgt, indem die Steuertaste CTRL gedrückt und festgehalten und dann der gewünschte Buchstabe gedrückt wird. Danach werden beide Tasten freigegeben.

Tabelle 1: Steuerzeichen

| Kommando | Bedeutung |
|---|---|
| CTRL/E | der Kursor wird auf den Anfang der nächsten Zeile gestellt, ohne die vorherige Eingabe zu löschen |
| CTRL/H | der Kursor wird um eine Position nach links bewegt, ein Zeichen wird gelöscht - gleiche Wirkung wie Rücksetztaste |
| CTRL/I | der Kursor bewegt sich zum nächsten Tabulatorhalt, Tabulatorhalts werden automatisch nach jeweils acht Spalten gesetzt - gleiche Wirkung wie TAB-Taste |
| CTRL/J | der Kursor wird an den Anfang der soeben eingegebenen Zeile bewegt, das eingegebene Kommando wird an SCP übergeben - gleiche Wirkung wie RETURN-Taste |
| CTRL/M | gleiche Wirkung wie CTRL/J |
| CTRL/R | auf die aktuelle Kursorposition wird ein # geschrieben, der Kursor wird auf die nächste Zeile bewegt und der bis dahin eingegebene Teil des Kommandos wird wiederholt |
| CTRL/U | die Zeichen der soeben eingegebenen Kommandozeile werden gelöscht, auf die aktuelle Kursorposition wird ein # geschrieben und der Kursor wird auf die nächste Zeile positioniert |
| CTRL/X | alle Zeichen der soeben eingegebenen Zeile werden gelöscht und der Kursor auf den Anfang der aktuellen Zeile positioniert |

Einige Steuerzeichen haben die gleiche Bedeutung, z.B. haben CTRL/J und CTRL/M die gleiche Wirkung wie die Rücksetz-Taste (backspace). Alle drei übergeben die Kommandozeile an SCP zur Ausführung. CTRL/H hat die gleiche Wirkung wie die Rücksetz-Taste.

## 1.4. Dateisicherung

Sowohl Bedienfehler als auch Gerätefehler können zur Zerstörung von Programmen oder Datendateien führen. Z.B. kann ein Fehler bei der Kommandoeingabe das Löschen eines gerade erzeugten Programmes bewirken. Ähnliche Folgen haben technische Fehler. Erfahrene Nutzer schützen sich vor dem Verlust von Programmen und Daten durch die fortlaufende Anfertigung von Arbeitskopien. Die Originale werden gesichert. Bei fehlerhafter Arbeitskopie kann jederzeit vom Original eine neue hergestellt werden. Während der laufenden Arbeit ist es sinnvoll, regelmäßig Kopien von neuen Programmen und Datendateien zu erstellen.
Üblicherweise werden Kopien dann erstellt, wenn bei Zerstörung einer Datei die Wiederherstellung der Informationen ca. 10 bis 20 mal länger dauert als das Kopieren.
Beim Kopieren gibt es zwei Möglichkeiten, entweder das Kopieren einer einzelnen Datei oder das Kopieren der kompletten Platte. Die Entscheidung ist abhängig davon, wieviel Dateien sich auf der Platte befinden. Das Kopieren einer Datei kann weniger als eine Minute dauern, aber zum Kopieren der ganzen Platte werden auch nur 2 bis 3 Minuten gebraucht.

## 1.5. Kopieren der SCP-Platte

Zum Sichern der SCP-Platte werden eine oder mehrere Platten für die Kopien, die Transientroutine COPYDISK und eine SCP-Originalplatte benötigt. Die Platten für die Kopien können fabrikneu oder gebraucht sein.
Verschiedene Disketten haben eine Schreibschutzmarkierung. Soll eine solche Diskette beschrieben werden, so muß diese Markierung mit einer speziellen Folie überklebt werden.
Mit dem zum Rechner gehörenden Formatierprogramm müssen die zum Kopieren vorgesehenen Platten entweder neu oder reformatiert werden,
Die gebrauchten Platten sind stets zu prüfen, ob sie Informationen enthalten, die noch benötigt werden.
COPYDISK kopiert eine Quellplatte vollständig auf eine Zielplatte, einschließlich gelöschter Bereiche, und überschreibt alle vorher gespeicherten Informationen. Auch zum Kopieren einer SCP-Originalplatte wird die Routine COPYDISK verwendet. Dazu wird die Systemplatte in das Laufwerk A und eine formatierte Platte in das Laufwerk B eingelegt, das System gestartet und nach seiner Bereitmeldung das Kommando

```
A>COPYDISK
```

eingegeben. SCP lädt COPYDISK in den Speicher und startet es.

Auf dem Bildschirm erscheinen die folgenden Ausschriften:

```
COPYDISK V 1.0
Source disk? A

Copy SCP1700 disk (N/<CR>)? <CR>

Destination disk? B

Format destination disk (Y/N)? N

Insert source disk in drive A
Insert destination disk in drive B
Type <CR> to continue <CR>

Reading Track 0...
Copy completed
Copy another disk (Y/N) ?N

Copy programm exiting
A>
```

Damit befindet sich eine exakte Kopie des Original-SCP auf der Platte im Laufwerk B, die Originalplatte kann an einem sicheren Ort aufbewahrt werden.
Von dieser gesicherten und unveränderten Originalplatte können beliebige Arbeitskopien angefertigt werden.
Nachdem die Arbeitskopie in das Laufwerk A eingelegt ist, kann sie als Systemplatte verwendet und mit ihr weitergearbeitet werden, z.B. können auch von ihr weitere Arbeitskopien des Systems erstellt werden.

## 2. Dateien, Platten, Laufwerke und Geräte

Die Hauptaufgabe des SCP ist der Zugriff zu Dateien und deren Wartung auf der Platte. SCP kann Programm- und Datendateien erstellen, lesen, schreiben, kopieren und löschen.

In diesem Kapitel wird erklärt, was eine Datei ist, wie sie erstellt, bezeichnet, zu ihr zugegriffen wird und wie Dateien auf der Platte gespeichert werden. Außerdem wird erläutert, wie ein Plattenwechsel und eine neue Zuordnung eines Standardlaufwerkes vorzunehmen sind.

### 2.1. Dateien

Eine SCP-Datei ist eine Sammlung von Informationen, die zueinander in Beziehung stehen und auf einer Platte gespeichert sind. Jede Datei muß einen eigenen Namen besitzen, weil über diesen Namen der Zugriff zur Datei erfolgt. Auf jeder Platte wird außerdem ein Dateiverzeichnis (directory) geführt. Dieses Verzeichnis enthält eine Liste aller auf der Platte gespeicherten Dateien und deren Plätze auf der Platte.

Im allgemeinen gibt es zwei Arten von Dateien: Programmdateien und Datendateien. Eine Programmdatei ist eine ausführbare Datei, eine Folge von Befehlen, die der Rechner schrittweise ausführen kann. Eine Datendatei ist eine Sammlung von Informationen, wie z.B. eine Liste von Namen und Adressen, ein Inventarverzeichnis, der Text eines Dokumentes, die Buchführung eines Geschäftsvorganges oder ähnliche voneinander abhängige Informationen. Der Rechner kann z.B. Namen und Adressen nicht ausführen, aber er kann ein Programm ausführen, das diese Namen und Adressen auf Briefe schreibt.

Eine Datendatei kann auch den Quellcode eines Programms enthalten. Im allgemeinen muß eine Quelldatei erst von einem Assembler oder Compiler übersetzt werden, ehe sie eine ausführbare Datei wird. In den meisten Fällen bearbeitet ein ausführbares Programm eine Datendatei. Jedoch gibt es auch Fälle, wo ein ausführbares Programm ein anderes ausführbares Programm bearbeitet, z.B. kann das ausführbare Kopierprogramm PIP Programmdateien kopieren.

### 2.2. Erstellen von Dateien

Es gibt viele Möglichkeiten, Dateien zu erzeugen. Eine Datei kann erstellt werden, indem eine vorhandene Datei auf einen neuen Platz kopiert und umbenannt wird. Unter SCP kann das Transientprogramm PIP zum Kopieren und Umbenennen benutzt werden. Ein anderer Weg, eine Datei zu erstellen, ist die Anwendung des Texteditors. Der SCP-Editor ED kann eine Datei erstellen und ihr den gewünschten Namen zuweisen. Schliesslich erstellen einige Programme, wie z.B. der Assembler, Ausgabedateien, so wie sie Eingabedateien verarbeiten.

## 2.3. Benennung von Dateien

SCP identifiziert jede Datei durch ihre eigene Dateispezifikation. Eine Dateispezifikation kann drei Teile haben:

| | | | |
|---|---|---|---|
| d: | Laufwerkskennung | ein Buchstabe | wahlfrei |
| filename | Dateiname | 1 - 8 Zeichen | |
| typ | Dateityp | 0 - 3 Zeichen | wahlfrei |

Eine Dateispezifikation besteht aus Buchstaben und Zahlen.
Da der Kommandoprozessor des SCP die folgenden Spezialzeichen als Trennzeichen behandelt, dürfen diese nicht innerhalb eines Dateinamens oder Dateityps verwendet werden.

```
< > . , ; : = ? * [ ]
```

Eine Dateispezifikation kann z.B. einfach ein Dateiname sein, der aus 1 - 8 Zeichen besteht:

```
MYFILE1
```

Ein Dateiname sollte so gewählt werden, daß er Hinweise zum Inhalt der Datei gibt. Enthält eine Datei z.B. eine Kundenliste, so könnte sie

```
CUSTOMER
```

heißen.

Sinnvollerweise werden bei der Arbeit mit SCP Dateien entsprechend ihrer Verwendung in Familien zusammengefaßt. Den SCP-Dateien werden aus ein bis drei Zeichen bestehende Familiennamen zugeordnet, die Dateitypen. Dateiname und Dateityp sind durch einen Punkt zu trennen. Wie bei den Dateinamen über den Inhalt sollten auch bei den Dateitypen die gewählten Zeichen Auskunft über die Dateifamilie geben. Im oben verwendeten Beispiel könnte der Kundendatei folgender Typ zugeordnet werden:

```
CUSTOMER.NAM
```

Wenn SCP Dateispezifikationen als Antwort auf ein DIR-Kommando anzeigt, dann füllt es kurze Dateinamen durch Leerzeichen auf, so daß die Dateinamen untereinander stehen und einfach vergleichbar sind. Die ausführbaren Programmdateien, die SCP von der Platte in den Speicher lädt, haben verschiedene Dateinamen, aber gehören zur Familie der Programme, die unter SCP laufen. Der Dateityp CMD identifiziert diese Familie als ausführbare Programme.

SCP legt verschiedene Dateifamilien fest. Die folgende Tabelle zeigt einige dieser Dateitypen mit einer kurzen Beschreibung jeder Familie.

Tabelle 2: SCP-Dateitypen

| Dateityp | Bedeutung |
|---|---|
| CMD | Maschinencodeprogramm |
| BAS | BASIC-Quellprogramm |
| $$$ | temporäre Datei |
| A86 | ASM86-Quelldatei |
| H86 | assemblierte A86-Datei im hexadezimalen Format |
| SUB | Liste von Kommandos, die mit SUBMIT ausgeführt werden |

## 2.4. Zugriff zu Dateien

Wird in einem Kommando eine Dateispezifikation angegeben, so sucht das jeweils zugeordnete residente oder transiente Kommando diese Datei auf der Platte, die sich im angezeigten Laufwerk befindet.

Beispiel:

```
A>dir copydisk.cmd
```

SCP durchsucht das Inhaltsverzeichnis der Platte im Laufwerk A nach COPYDISK.CMD. Soll auf ein anderes Laufwerk zugegriffen werden, z.B. auf B, so ist die Laufwerkskennung gefolgt von einem Doppelpunkt vor dem Dateinamen anzugeben:

```
A>dir b:myfile.lib
```

SCP sucht die Datei MYFILE.LIB im Inhaltsverzeichnis der Platte im Laufwerk B.

Eine Laufwerkskennung kann auch vor ein ausführbares Programm geschrieben werden:

```
A>b:pip
```

SCP sucht im Verzeichnis der Platte im Laufwerk B nach der Datei PIP.CMD. Wird PIP dort gefunden, so lädt es SCP in den Speicher und startet es.

## 2.5. Zugriff zu mehreren Dateien

Einige residente und transiente Kommandos des SCP können Dateien auswählen und bearbeiten, die Dateigruppensymbole (wildcard) im Dateinamen oder Datentyp enthalten.
Eine Dateispezifikation, die diese Sonderzeichen enthält, kann sich auf mehrere Dateien beziehen. Die jeweiligen Dateispezifikationen stellen wegen der Sonderzeichen für SCP ein Vergleichsmuster dar. SCP sucht im Plattenverzeichnis nach den Dateien, die im Vergleichsmuster übereinstimmen.

Benutzt werden die beiden Sonderzeichen ? und *. Für den Gebrauch dieser Sonderzeichen gelten folgende Regeln:

- Ein ? kennzeichnet eine Position, die nicht auf Übereinstimmung geprüft wird (einschließlich Leerzeichen).
- Ein * muß das letzte oder einzige Zeichen in einem speziellen Dateinamen bzw. -typ sein. SCP ersetzt intern das *-Zeichen durch ?-Zeichen bis zum Ende des speziellen Dateinamens oder Dateityps.
- Wenn der zu vergleichende Dateiname kürzer als 8 Zeichen ist, wird er von SCP weiter mit Leerzeichen aufgefüllt.
- Wenn der zu vergleichende Dateityp kürzer als 3 Zeichen ist, wird er vom SCP weiter mit Leerzeichen aufgefüllt.

Befinden sich auf einer Platte die folgenden 6 Dateien A.CMD, AA.CMD, AAA.CMD, B.CMD, A.A86 und B.A86, dann ergeben sich durch Anwendung der Sonderzeichen folgende Varianten:

| | |
|---|---|
| *.* | wird behandelt wie ????????.??? |
| ????????.??? | bezieht sich auf alle 6 Dateien |
| *.CMD | wird behandelt wie ????????.CMD |
| ????????.CMD | bezieht sich auf alle Dateien vom Typ CMD |
| ?.CMD | bezieht sich auf A.CMD und B.CMD |
| ?.* | wird behandelt wie ?.??? |
| ?.??? | bezieht sich auf A.CMD, B.CMD, A.A86 und B.A86 |
| A?.CMD | bezieht sich auf A.CMD und AA.CMD |
| A*.CMD | wird behandelt wie A???????.CMD |
| A???????.CMD | bezieht sich auf A.CMD, AA.CMD und AAA.CMD |

Zu beachten ist, daß SCP Sonderzeichenmuster nur beim Durchsuchen von Plattenverzeichnissen benutzt und diese nur in speziellen Dateinamen und Dateitypen zulässig sind. Ein Sonderzeichen kann nicht in einer Laufwerkskennung benutzt werden.

## 2.6. Organisation und Schutz von Dateien

Unter SCP können Dateien in Gruppen organisiert, vor zufälligem Zugriff geschützt und so spezifiziert werden, wie sie vom DIR-Kommando anzuzeigen sind. SCP unterstützt diese Eigenschaften durch Zuweisung von Nutzernummern und Attributen zu den Dateien. Die Eigenschaften werden im Plattenverzeichnis vermerkt.

Mit Nutzernummern können Dateien in 16 Dateigruppen unterteilt werden. Alle Dateien können durch Nutzernummern identifiziert werden, die im Bereich von 0 bis 15 liegen. SCP weist einer Datei bei ihrer Erstellung eine Nutzernummer zu. Wenn nicht das Kommando PIP zum Kopieren einer Datei mit Zuordnung einer anderen Nutzernummer zur kopierten Datei benutzt wird, erhält die Datei die aktuelle Nutzernummer. Zum Anzeigen und Verändern der aktuellen Nutzernummer kann das residente Kommando USER benutzt werden.
Die meisten Kommandos können nur zu den Dateien zugreifen, die die aktuelle Nutzernummer haben. Ist z.B. die aktuelle Nutzernummer 7, dann zeigt ein DIR-Kommando nur die Dateien an, die unter dieser Nutzernummer erstellt wurden. Eine Ausnahme ist das PIP-Kommando. Mit wahlweisen Angaben [Gn] kopiert PIP eine Datei mit einer Nutzernummer und gibt der Kopie eine andere Nutzernummer (siehe Abschnitt 4.13.).

Dateiattribute steuern den Dateizugriff. Es gibt zwei Arten von Dateizugriffsattributen. Das Attribut DIR/SYS kann entweder auf DIR (Verzeichnis) oder SYS (System) gesetzt werden. Wenn eine Datei erstellt wird, wird sie automatisch mit dem DIR-Attribut gekennzeichnet. Das DIR-Kommando zeigt nur Dateien an, die im aktuellen Nutzerbereich sind.

Die Zuweisung des Attributes DIR/SYS zu einer Datei erfolgt mit dem transienten Kommando STAT. Das DIR-Kommando zeigt keine Dateien an, die mit dem Attribut SYS versehen sind. Um diese Dateien anzuzeigen, muß das Kommando DIRS benutzt werden. Dabei ist zu beachten, daß DIRS ebenfalls nur die Systemdateien anzeigt, die sich in aktuellen Nutzerbereich befinden. Das STAT-Kommando zeigt auch Dateien mit dem SYS-Attribut an, aber ebenfalls nur aus dem aktuellen Nutzerbereich.

Insbesondere ist es sinnvoll, den Dateien SYS-Attribute zuzuweisen, die sich im Nutzerbereich 0 befinden. Sie sollten ausführbare Kommandodateien mit dem Typ CMD sein. Wenn einer Kommandodatei mit Nutzernummer 0 ein SYS-Attribut zugewiesen ist, kann diese Datei von jedem beliebigen Nutzerbereich auf dem gleichen Laufwerk aus gelesen und ausgeführt werden. Diese Eigenschaft hat praktische Bedeutung, wenn allgemein nutzbare Programme für jeden Nutzerbereich verfügbar sein sollen, ohne eine Kopie dieser Programme in jedem Nutzerbereich mitzuführen.

Das RW/RO-Attribut kann entweder auf RO (Read-Only/Nur Lesen) oder RW (Read-Write/Lesen und Schreiben) gesetzt werden. Eine Datei mit dem Zugriffsattribut RW kann gelesen und geschrieben werden, außer wenn die Platte schreibgeschützt ist (hard ware) oder das Laufwerk eine Platte enthält, die auf Read-Only gesetzt ist (soft ware).
Ist eine Datei mit RO markiert, dann führt jeder Schreibzugriff zu dieser Datei zu einer Read-Only-Fehlermitteilung. Wichtige Dateien können mit dem RO-Attribut geschützt werden.

Das RW/RO-Attribut kann einer Datei oder einer Dateigruppe mit dem STAT-Kommando zugewiesen werden. STAT kann das RO-Attribut auch einem Laufwerk zuweisen. Das Steuerkommando CTRL/C setzt alle eingegliederten Laufwerke auf das RW-Attribut.

## 2.7. Dateispeicherung auf der Platte

SCP führt den Dateinamen, den Dateityp, die Nutzernummer und Attribute jeder Datei in einem speziellen Bereich auf der Platte, dem Plattenverzeichnis. In dieses Verzeichnis wird von SCP auch eingetragen, welche Plattensektoren zu welcher Datei gehören. Das Verzeichnis ist so groß, daß bis zu 64 Dateien eingetragen werden können. Beim Löschen einer Datei wird Speicherbereich auf zwei Wegen wiedergewonnen: SCP stellt den Verzeichnisplatz zur Eintragung einér anderen Datei zur Verfügung und gibt den von der gelöschten Datei belegten Speicherbereich wieder frei für eine spätere Benutzung. Durch diese Eigenschaft der dynamischen Belegung wird das SCP sehr effektiv.

Weil SCP automatisch mehr Platz als notwendig zuweist und diesen für eine Wiederbelegung wieder freigibt, falls die Datei gelöscht wird, braucht dem SCP nicht mitgeteilt zu werden, wie groß eine Datei werden wird.

## 2.8. Plattenwechsel

Damit SCP mit einer Datei arbeiten kann, muß die Platte mit dieser Datei in einem Laufwerk sein, das sich im Bereitstatus befindet.
SCP kann dann auch zum Platteninhaltsverzeichnis zugreifen.
Plattenwechsel darf nur ausgeführt werden, wenn am Bildschirm die Eingabeanforderung des SCP angezeigt ist. Der ausgeführte Plattenwechsel ist dem SCP durch Eingabe von CTRL/C nach der Eingabeanforderung mitzuteilen. SCP veranlaßt für die neue Platte das Rücksetzen des Laufwerkes.

Wird nach Plattenwechsel die Eingabe von CTRL/C vergessen, schützt SCP automatisch die neue Platte. Beim Versuch, auf die neue Platte zu schreiben, z.B. mit dem Texteditor oder dem Kopierprogramm, bemerkt SCP, daß sich nicht mehr die Orginalplatte im Laufwerk befindet und bringt folgende Ausschrift:

```
Bdos err on d: RO
```

d: ist die Spezifizierung des Laufwerkes mit der neuen Platte. Nach Ausgabe dieser Fehlermeldung muß CTRL/C gegeben werden, um die Systemeingabeanforderung zu veranlassen und ein weiteres CTRL/C, um die neue Platte einzugliedern.

## 2.9. Wechsel des Standardlaufwerkes

Während SCP arbeitet, wird ein festgelegtes Laufwerk als aktuelles oder Standardlaufwerk bezeichnet.
SCP und die Dienstprogramme suchen Programme oder Datendateien immer auf der Platte im Standardlaufwerk, wenn nicht vom Bediener im Kommando eine andere Laufwerkspezifikation angegeben wurde. Das Standardlaufwerk ist aus der Meldung ablesbar.

```
A>
```

gibt an, daß das Laufwerk A das Standardlaufwerk ist. Bei der SCP-Kommandoeingabe muß immer das Standardlaufwerk beachtet werden. Wird SCP gestartet, ist gewöhnlich das Laufwerk A das Standardlaufwerk. Stehen mehrere Laufwerke zur Verfügung, kann das Standardlaufwerk ausgewechselt werden. Das geschieht durch Eingabe der Laufwerkspezifikation für das Laufwerk, auf das gewechselt werden soll.

```
A>B:
```

Daraufhin wird das Laufwerk B zum Standardlaufwerk erklärt. Als Meldung erscheint dann:

```
B>
```

Danach suchen SCP und die Dienstprogramme immer auf der Platte im Laufwerk B nach Programmen oder Datendateien, sofern keine andere Laufwerkspezifikation im Kommando angegeben wird.

## 2.10. Zugriffsstatus für Laufwerke

Unter SCP können Laufwerke ebenso wie Dateien mit RO-Attributen gekennzeichnet werden. Der Standardstatus eines Laufwerkes ist RW, aber SCP markiert das Laufwerk automatisch mit RO, wenn die Platte darin gewechselt wurde. Einem Laufwerk kann mit dem Transientprogramm STAT das RO-Attribut zugewiesen werden (siehe Abschnitt 4.15.). Um diese Zuweisung rückgängig zu machen, muß nach der Meldung CTRL/C eingegeben werden.

## 2.11. Weitere SCP-Geräte

SCP bedient alle am Rechner angeschlossenen Geräte. SCP nutzt logische Geräte. Die folgende Tabelle zeigt diese logischen Geräte und gibt an, ob sie Eingabe- oder Ausgabegeräte sind.

Tabelle 3: Logische Geräte des SCP

| Gerätename | Gerätetyp |
|---|---|
| CON: | Terminal - Eingabe und Ausgabe |
| AXI: | Hilfsgerät - Eingabe |
| AXO: | Hilfsgerät - Ausgabe |
| LST: | Listengerät - Ausgabe |

SCP verbindet physische Geräte mit den logischen Gerätenamen. Z.B. ist das Standardterminal für die Eingabe die Tastatur und das Standardterminal für die Ausgabe der Bildschirm. Wenn einem logischen ein anderes Peripheriegerät zugewiesen werden soll, wird dazu das STAT-Kommando benutzt. Z.B. können der Terminaleingabe ein Fernschreiber und dem LST:-Gerät ein Drucker zugewiesen werden.

Einem logischen Eingabe- bzw. Ausgabegerät kann jeweils nur ein physisches Eingabe- bzw. Ausgabegerät zugewiesen werden (siehe Abschnitt 4.15.).

## 3. SCP-Kommandokonzept

Wie im Abschnitt 1 beschrieben wurde, besteht das SCP-Kommando aus einem Kommandoschlüsselwort, einer wahlweisen Kommandoparameterfolge und dem Endezeichen RETURN.

In diesem Abschnitt werden alle Kommandos (residente und transiente) erläutert. Außerdem erfolgt eine Erklärung der Steuerzeichen des SCP.

### 3.1. Kommandotypen

Ein Kommandoschlüsselwort identifiziert ein Programm, das sich entweder im Speicher als Teil des SCP oder als Programmdatei auf der Platte befindet. Ist das identifizierte Programm Bestandteil des SCP im Speicher, so heißt es residentes Kommando. Ein Kommando als Programmdatei auf der Platte wird als transientes Kommando oder einfach als Dienstprogramm bezeichnet.
Den transienten Kommandos können nach Belieben weitere hinzugefügt werden.

### 3.2. Residente Kommandos

Residente Kommandos sind Teil des SCP und befinden sich immer im Speicher, unabhängig davon, welche Platte sich in welchem Laufwerk befindet. Sie werden schneller ausgeführt als transiente Kommandos. Die residenten Kommandos sind ausführlich im Abschnitt 4. beschrieben, die folgende Tabelle enthält eine Zusammenstellung.

Tabelle 4: Residente Kommandos

| Kommando | Bedeutung |
|---|---|
| DIR | listet die Namen der Datei mit dem DIR-Attribut aus einem Plattenverzeichnis tabellarisch aus |
| DIRS | listet die Namen der Datei mit dem SYS-Attribut aus einem Plattenverzeichnis tabellarisch aus |
| ERA | löscht einen Dateinamen aus einem Plattenverzeichnis und gibt den von dieser Datei belegten Speicherplatz auf der Platte wieder frei |
| REN | dient zum Umbenennen von Dateien im Plattenverzeichnis |
| TYPE | gibt den Inhalt einer Textdatei auf dem Bildschirm aus |
| USER | dient zum Ändern der Nutzernummer |

## 3.3. Transiente Dienstprogramme

Ein transientes Kommando gelangt nur in den Speicher, wenn es angefordert wird. Im Abschnitt 4. werden die in der folgenden Tabelle zusammengefaßten transienten Kommandos ausführlich beschrieben (ED im Abschnitt 5.).

Tabelle 5: SCP-Dienstprogramme

| Kommando | Bedeutung |
|---|---|
| ASM86 | übersetzt Assemblerprogramme in die Maschinencodeform |
| COPYDISK | kopiert eine Platte, die das SCP, Programm- und Datendateien enthalten kann |
| DDT86 | dient zum Test von Programmen und zur interaktiven Korrektur von Programmfehlern |
| DISKSET | ändert die Zugriffsparameter zu einem Plattenlaufwerk |
| ED | erstellt oder ändert Textdateien |
| GENCMD | erzeugt aus der Ausgabedatei des ASM86 eine ausführbare Kommandodatei |
| HELP | zeigt auf den Bildschirm an, wie jedes SCP-Kommando genutzt werden kann |
| INIT | formatiert eine Platte und initialisiert den SCP-Verzeichnisbereich |
| LDCOPY | kopiert den SCP-1700-Lader |
| PIP | fügt und kopiert Dateien |
| STAT | dient zum Prüfen und Ändern von Statusinformationen und zur Zuweisung physischer E/A-Geräte zu logischen SCP-Geräten |
| SUBMIT | sendet eine Datei, die Kommandos enthält, zur Ausführung an das SCP |

## 3.4. SCP-Kommandoauswertung

Falls ein Kommandoschlüsselwort kein residentes Kommando identifiziert, sucht SCP auf dem Standard- bzw. angegebenen Laufwerk nach der Programmdatei, die den im Schlüsselwort angegebenen Dateinamen und den Typ CMD hat.

Beispiel:

```
A>ED MYPROG.BAS
```

SCP führt dieses Kommando in folgenden Schritten durch:

1) SCP findet zuerst das Schlüsselwort ED, das kein residentes Kommando identifiziert.
2) SCP sucht die Datei ED.CMD auf der Platte im Standardlaufwerk. Findet es diese Datei nicht unter der aktuellen Nutzernummer, so sucht es sie unter der Nummer 0 mit dem SYS-Attribut.
3) Findet SCP die Datei, wird diese in den Speicher gelesen und ED gestartet.
4) ED behält die Steuerung, bis ein Abbruchkommando eingegeben wird.
5) SCP gibt die Systemmeldung aus und wartet auf die Eingabe eines weiteren Kommandos.

Wenn SCP weder ein residentes noch ein transientes Kommando entsprechend dem Kommandoschlüsselwort findet, erfolgt eine Fehlerausschrift, die das eingegebene Schlüsselwort gefolgt von einem Fragezeichen enthält. Die Ursache können folgende Fehler sein:

- Das Schlüsselwort ist weder ein residentes noch ein transientes Kommando.
- Es gibt keine zugehörige CMD-Datei, weder unter der aktuellen Nutzernummer noch unter der Nummer 0 mit dem Attribut SYS.
- Auf einem speziell angeforderten Laufwerk gibt es keine Platte, die die zugehörige CMD-Datei im aktuellen Nutzerbereich oder mit SYS-Attribut unter Nutzernummer 0 enthält.

Enthält die Standardplatte nur die transienten SCP-Kommandos und erfolgt die Eingabe:

```
A>EDIT MYPROG.BAS
```

so durchläuft SCP folgende Schritte bis zur Fehlermitteilung:

1) SCP findet unter EDIT kein residentes Kommando.
2) SCP findet auf der Platte im Standardlaufwerk weder unter der aktuellen noch unter der Nutzernummer 0 die Datei EDIT.CMD.
3) SCP bringt die Ausschrift:

   ```
   EDIT?
   ```

   Das bedeutet, dieses Kommando kann nicht ausgeführt werden.
4) Danach erfolgt die Ausgabe der Systemmeldung, SCP wartet auf die nächste Kommandoeingabe.

## 3.5. Steuerzeichenkommandos

SCP kann durch Drücken von Spezialtasten veranlaßt werden, bestimmte Funktionen auszuführen. Durch Nutzung der CTRL-Tastenkommandos wird z.B. die Rollfunktion des Bildschirmes ein- oder ausgeschaltet, die laufende Operation abgebrochen oder das Echo der Bildschirmausgabe auf den Drucker gelegt. In der folgenden Tabelle sind die Steuerzeichenkommandos zusammengefaßt.

Tabelle 6: Steuerzeichenkommandos

| Kommando | Bedeutung |
|---|---|
| CTRL/C | beendet die Arbeit des laufenden Programms, oder wenn die Eingabe der Systemmeldung erfolgt, initialisiert es das System und das Standardlaufwerk und setzt alle Laufwerke auf den RW-Status. |
| CTRL/P | wiederholt sämtliche Bildschirmausgaben, auf dem Drucker (Echo). Ein zweites CTRL/P beendet das Echo. Die Wirkung tritt nur ein, wenn das System einen Drucker besitzt. |
| CTRL/S | unterbricht das Rollen des Bildes auf dem Bildschirm. Rollt das Bild zu schnell, kann CTRL/S gedrückt werden. Erneutes Drücken von CTRL/S oder auch einer beliebigen anderen Taste bewirkt die Fortsetzung der Anzeige. |

## 4. Kommandoübersicht

In diesem Abschnitt werden residente und transiente Kommandos vermischt beschrieben. Jedes Kommando wird in allgemeiner Form (Syntax) angegeben und durch Beispiele erweitert. Komplizierte Kommandos werden später beschrieben, ED z.B. im Abschnitt 5., die Dienstprogramme ASM86 und DDT86 in der Anleitung für den Programmierer und GENCMD in der Anleitung für den Systemprogrammierer.

### 4.1. Formale Bedingungen

Wie bereits vorgestellt, gibt es drei verschiedene Teile in einer Dateispezifikation. Um die Erläuterung zu vereinfachen, erhalten die drei Teile formale Namen.

Die drei Teile der Dateispezifikation sind Laufwerkskennung (driver specifier), Dateiname (filename) und Dateityp (filetyp).

| | |
|---|---|
| Laufwerkskennung | die wahlweise Angabe des Plattenlaufwerkes, das die Platte mit der Datei oder der Gruppe von Dateien enthält, auf die zugegriffen werden soll. Der Laufwerkskennung muß ein Doppelpunkt folgen. |
| Dateiname | eine bis zu acht Zeichen lange Bezeichnung einer Datei oder Dateigruppe. |
| Dateityp | der wahlweise bis zu drei Zeichen lange Familienname einer Datei oder Dateigruppe. Wird ein Dateityp angegeben, muß er durch einen Punkt vom Dateinamen getrennt werden. |

Die allgemeine Form einer Dateispezifikation ist:

```
d:filename.typ
```

In dieser Form entspricht d: der wahlweise angebbaren Laufwerkskennung, filename repräsentiert den ein bis acht Zeichen langen Dateinamen und .typ die wahlweise Angabe des ein bis drei Zeichen langen Dateityps.

Zulässige Kombinationen der Elemente einer SCP-Dateispezifikation sind:

- filename
- d:filename
- filename.typ
- d:filename.typ

Wird keine Laufwerkskennung angegeben, benutzt SCP das Standardlaufwerk. Fehlt der Punkt und der Dateityp, führt SCP drei Leerzeichen als Dateityp ein. Diese allgemeine Form heißt Dateispezifikation.
Eine Dateispezifikation bezeichnet eine bestimmte Datei oder eine Gruppe von Dateien im Platteninhaltsverzeichnis, das durch die Laufwerkskennung spezifiziert ist, z.B. ist

```
B:MYFILE.A86
```

eine Dateispezifikation, die das Laufwerk B:, den Dateinamen MYFILE und den Dateityp A86 spezifiziert. Zur Abkürzung wird einfach filespec in der Beschreibung der Kommandosyntax benutzt.

Einige SCP-Kommandos akzeptieren spezielle Sonderzeichen (Dateigruppensymbole) im Dateinamen und Dateityp, z.B. ist

B:MY*.A??

eine Dateispezifikation mit der Laufwerkskennung B:, dem Dateinamen MY* und dem Dateityp A??. Diese Dateispezifikation kann mit mehreren Dateien im Verzeichnis übereinstimmen.

## 4.2. Beschreibungsform der Kommandos

Dieser Abschnitt enthält die Beschreibung der residenten und transienten Kommandos. Jede Kommandobeschreibung erfolgt in spezifischer Form:

- Die Beschreibung beginnt mit dem Kommandoschlüsselwort in Großbuchstaben. Ist es sinnvoll, Kommentare zur effektiven Beschreibung der Möglichkeiten zu benutzen, werden diese in runde Klammern gesetzt.

- Unter Syntax werden eine oder mehrere allgemeine Formen des Aufbaus der Kommandozeile angegeben.

- Der Typ zeigt, ob das Schlüsselwort ein residentes oder transientes Kommando bezeichnet. Residente Kommandos sind immer nutzbar, während transiente Kommandos das zugehörige Programm von einer angeschlossenen Platte anfordern.

- Unter Anwendung wird der allgemeine Gebrauch des Kommandoschlüsselwortes definiert.

- Die Bemerkungen geben Ausnahmen und Spezialfälle an.

- Als Beispiele werden zulässige Kommandos für das entsprechende Schlüsselwort angegeben. Um Bedienereingaben von Systemausschriften zu unterscheiden, werden erstere unterstrichen. Auf dem Bildschirm entfallen diese Unterscheidungen.

Die Darstellung in den Syntaxzeilen beschreibt die allgemeine Kommandoform nach folgenden Regeln:

- Worte in Großbuchstaben müssen vom Bediener eingegeben werden. Dabei können Kombinationen von Groß- und Kleinbuchstaben benutzt werden.

- Ein Wort in Kleinbuchstaben hat allgemeine Bedeutung, die im Bescheibungstext für das Kommando erläutert ist. Wenn z.B. das Wort wahlweise angegeben ist, kann aus einer Liste von Angaben ausgewählt werden.

- n kann durch eine Zahl ersetzt werden.

- Die symbolischen Bezeichnungen d:, filename, .typ und filespec haben die allgemeine bereits beschriebene Bedeutung.

- Wenn ein Leerzeichen angegeben ist, müssen ein oder mehrere Leerzeichen eingesetzt werden, außer wenn eine andere Festlegung getroffen wurde. Z.B. brauchen die wahlweisen PIP-Angaben nicht durch Leerzeichen getrennt werden.

- Angaben in geschweiften Klammern { } sind wahlweise. Das Kommando kann ohne diese Angaben benutzt werden.

- Drei Punkte (...) geben an, daß der vorhergehende Ausdruck beliebig oft wiederholt werden kann.

- Alternativen werden durch einen senkrechten Strich | getrennt. Der | kann als oder gelesen werden.

- Das Zeichen ^ oder CTRL bezieht sich auf die CTRL-Taste der Tastatur.

- Alle anderen Angaben müssen im Kommando angegeben werden.

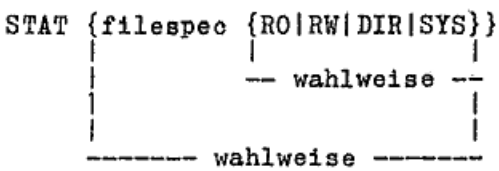
Beispiele:

Das STAT(Status)-Kommando zeigt den freien Speicherraum von angeschlossenen Platten und den von den einzelnen Dateien belegten Speicherraum in Kilobytes an.

Mit STAT können auch die Attribute RO, RW, SYS und DIR zugewiesen werden.

Die Syntax für das STAT-Kommando hat folgendes Aussehen:

Syntax:




Die Parameterangaben sind wahlweise. STAT allein ist ein zulässiges Kommando. STAT mit einer Dateispezifikation ist ebenfalls zulässig, auch STAT mit einer Dateispezifikation und einem der Attribute RO, RW, DIR oder SYS.
Demzufolge ist

```
STAT filespec
```

ein zulässiges Kommando.

Entsprechend der Syntax können verschiedene Kommandozeilen aufgebaut werden:

```
STAT
----
STAT X.A86
----------
STAT X.A86 RO
-------------
STAT X.A86 SYS
--------------
STAT *.A86
----------
STAT *.* RW
-----------
STAT X.* DIR
------------
```

Das SCP-Kommando PIP (Dateitransferprogramm) ist ein Kopierprogramm für Dateien. PIP kann Informationen vom Bildschirm auf die Platte oder den Drucker kopieren. PIP kann zwei oder mehr Dateien zu einer längeren Datei kombinieren. PIP kann Dateien nach dem Kopieren umbenennen.

Syntax:

```
PIP dest-filespec=source-filespec{,filespec...}
```

In diesem Beispiel ist dest-filespec die Dateispezifikation des Zieles oder ein peripheres Gerät (z.B. Drucker), das die Daten empfängt. Analog ist source-filespec die Dateispezifikation der Quelle oder ein peripheres Gerät (z.B. Tastatur), das die Daten überträgt. PIP akzeptiert Dateigruppensymbole im Dateinamen und Datentyp. Aus dieser Syntax lassen sich viele verschiedene Kommandozeilen ableiten.

Einige von ihnen sind folgende:

```
PIP NEWFILE.DAT=OLDFILE.DAT
---------------------------
PIP B:=A:THISFILE.DAT
---------------------
PIP B:X.BAS=Y.BAS,Z.BAS
-----------------------
PIP X.BAS=A.BAS,B.BAS,C.BAS
---------------------------
PIP B:=A:*.BAK
--------------
PIP B:=A:*.*
------------
```

## 4.3. Kommando ASM86 (Assembler)

Syntax:

ASM86 filespec {¤parameter-list}

Typ:

Transientprogramm

Anwendung:

Das ASM86-Dienstprogramm übersetzt Quellcode in den Maschinencode. Die Arbeit mit dem ASM86 wird in der Anleitung für den Programmierer ausführlich beschrieben.

filespec bezeichnet eine Quelltextdatei, die ein Programm in Assemblersprache enthält und die übersetzt werden soll. Wird die Angabe des Dateityps weggelassen, wird A86 angenommen. Der Assembler benutzt das in filespec angegebene Laufwerk für die Ausgabedateien, wenn nicht im Parameterteil etwas anderes angegeben wurde.

Die drei Ausgabedateien, die der Assembler erstellt, sind folgende:

LST enthält die aus der Quelle gebildete Übersetzungsliste
H86 enthält den Maschinencode im Hex-Format
SYM enthält alle im Programm definierten Symbole mit ihren relativen Adressen

Der Assembler weist diesen drei Dateien den gleichen Dateinamen zu, der für die Quelldatei spezifiziert wurde.

Der Übersetzungsvorgang kann durch wahlweise Angabe von Parametern in der parameter-list gesteuert werden. Jeder Parameter besteht aus einem einzelnen Buchstaben, dem ein weiterer Buchstabe für den Gerätenamen folgt. Die Parameter können durch Leerzeichen getrennt werden, aber jedem Parameter muß der Buchstabe für den Gerätenamen unmittelbar folgen. Die Parameter sind A, H, P und S. Die Gerätenamen werden durch Buchstaben von A bis P (entsprechend dem gewünschten Laufwerk) angegeben. Die Buchstaben X, Y und Z haben eine besondere Bedeutung, wenn sie als Gerätenamen genutzt werden:

X bezieht sich auf den Bildschirm
Y bezieht sich auf den Drucker
Z unterdrückt die Ausgabe (Nullgerät)

Der A-Parameter wird benutzt, um für den Zugriff zur Quelldatei ein anderes als das Standardlaufwerk zuzuweisen. Zulässige Parameter sind AA bis AP.

Mit dem H-Parameter wird das Laufwerk für die H86-Datei zugewiesen. Zulässige Parameter sind HA bis HP, HX, HY, HZ.

Mit dem P-Parameter wird das Laufwerk für die LST-Datei zugewiesen. Zulässige Parameter sind PA bis PP, PX, PY, PZ.

Mit dem S-Parameter wird das Laufwerk für die SYM-Datei zugewiesen. Zulässige Parameter sind SA bis SP, SX, SY, SZ.

Wenn sich Parameterangaben widersprechen, gilt der am weitesten rechts stehende Parameter.

Beispiele:

```
A>ASM86 X
```

ASM86.CMD muß sich im Laufwerk A befinden. Die Quelldatei X.A86 wird vom Laufwerk A gelesen und die Dateien X.LST, X.H86 und X.SYM werden nach Laufwerk A geschrieben.

```
B>ASM86 X.ABC ¤PX
```

ASM86.CMD muß sich im Laufwerk B befinden. Die Quelldatei X.ABC wird ebenfalls von B gelesen. Die Übersetzungsliste wird über den Bildschirm ausgegeben, die Dateien X.H86 und X.SYM werden nach B ausgegeben.

```
A>ASM86 MYPROG ¤AB PY HC
```

Die Quelldatei MYPROG.A86 wird von B gelesen, die Liste zum Drukker gesendet, die Datei MYPROG.H86 zum Laufwerk C und die Datei MYPROG.SYM nach A geschrieben.

```
A>B:ASM86 X ¤SZ
```

ASM86.CMD muß sich auf B befinden. X.A86 wird von A gelesen. Die Dateien X.LST und X.H86 werden nach A geschrieben. Eine Datei X.SYM wird nicht generiert.

## 4.4. COPYDISK-Kommando (Platten kopieren)

Syntax:

COPYDISK

Typ:

Transientes Kommando

Anwendung:

COPYDISK kopiert alle Informationen von einer Quellplatte auf eine Zielplatte, einschließlich der SCP-1700-Systemspuren, sofern sie auf der Quellplatte vorhanden sind. Die Zielplatte muß für die Aufnahme von Daten vorbereitet, d.h. formatiert sein. COPYDISK bietet die Möglichkeit, die Zielplatte vor dem Kopieren zu formatieren.
Quell- und Zieldatei dürfen keine fehlerhaften Spuren enthalten.

Nach dem Aufruf

```
COPYDISK
```

meldet sich das Kopierprogramm mit

```
COPYDISK Vx.y

Source disk?
```

und erwartet die Eingabe der Laufwerksbezeichnung (A-P) für die Quellplatte.
Alle Eingaben sind durch (CR) abzuschließen.
Danach fragt COPYDISK, ob die Quellplatte eine SCP-1700-Platte ist.

```
Using current drive parameters (N/<CR>)?
```

Ist die Antwort <CR>, entnimmt COPYDISK die für den Kopiervorgang notwendigen Plattenparameter den entsprechenden Gerätesteuertabellen.
Bei Eingabe von N erfragt COPYDISK nacheinander die benötigten Parameter:

```
Tracks /Disk    :    (Spurenzahl)
Sectors/Track   :    (Sektoren/Spur)
Bytes  /Sector  :    (Sektorgröße)
Track 0 handling?    (Behandlung von Spur 0)
```

Die Spuranzahl wird vom System selbst festgelegt (80 Spuren bei 5,25", 77 Spuren bei 8"). Die folgenden beiden Werte sind durch den Bediener einzugeben. Ist die Spur 0 abweichend von den übrigen Spuren formatiert (einfache Dichte, Sektorgröße = 128 Byte, 16 bzw. 26 Sektoren), dann ist die Frage 'Track 0 handling?' mit Y (Yes) , sonst mit N (No) zu beantworten.

Die nächste Abfrage von COPYDISK gilt der Laufwerksbezeichnung für die Zielplatte:

```
Destination disk?
```

Für Ziel- und Quellplatte kann das gleiche Laufwerk zugewiesen werden. Ziel- und Quellaufwerk müssen vom gleichen Typ sein (beide 5,25" oder 8").

Bevor der Kopiervorgang beginnt, fragt COPYDISK, ob die Zielplatte zu formatieren ist.

```
Format destination disk? (Y/N)
```

Die Antwort Y (Yes), wird durch COPYDISK abgefragt, in welcher Reihenfolge die Sektoren physisch innerhalb einer Spur angeordnet werden sollen (interleave). Standard ist interleave = 1 (Eingabe <CR>). Der Start des Formatiervorganges erfolgt nach Ausführung der Bedieneranweisung:

```
Insert destination disk in drive x
Type <CR> to continue
```

(Zielplatte in Laufwerk x einlegen und mit <CR> fortsetzen)

In analoger Weise fordert COPYDISK auch Bedienerhandlungen zum Start bzw. während des Kopiervorgangs an.
Der erfolgreiche Abschluß des Kopierens wird durch die Ausgabe

```
COPY completed
```

COPY another disk (Y/N)?

(Kopieren beendet; soll eine andere Platte kopiert werden?)

anzeigt.
Bei Eingabe von Y beginnt der Dialog wieder mit der Frage nach dem Laufwerk für die Quellplatte.

Fehler:

Falsche Laufwerksangaben bzw. die Angabe unverträglicher Laufwerke für Ziel- und Quellplatte führen zur Fehlermeldung:

Illegal diskette drive

Danach wird mit einer erneuten Eingabeanforderung fortgesetzt.

Fehler beim Plattenzugriff werden durch:

DISK ERROR : Retry(R) or Ignore(I) or Cancel(C)?

angezeigt.
Die Eingabe von R bewirkt die Wiederholung des Plattenzugriffs. Durch I wird der Lese-/Schreibfehler ignoriert, d.h., die betreffende Spur wird nicht bzw. verfälscht kopiert. Jede andere Eingabe führt zum Abbruch des Kopiervorganges.

## 4.5. Kommando DDT86 (Dynamische Testhilfe)

Syntax:

DDT86{filespec}

Typ:

Transientprogramm

Anwendung:

DDT86 ermöglicht den Test von Programmen, die im Maschinencode vorliegen.

DDT86-Kommandos bestehen aus einfachen Buchstaben. Diese werden zusammen mit ihren Parametern und wahlweisen Angaben in Tabelle 7 beschrieben. Die Kommandobuchstaben sind Großbuchstaben. Wahlweise Angaben sind in geschweifte Klammern gesetzt. Die Parameter in dieser Tabelle sind durch entsprechende Werte zu ersetzen. Eine Übersicht über die Bedeutung der einzelnen Parameter steht nach der Tabelle 7.

Tabelle 7: DDT86-Kommandos

| Kommando | Bedeutung |
|---|---|
| As | Eingabe von Assemblerbefehlen |
| Bs,f,s1 | Vergleich von Speicherblöcken |
| D{W}{s{,f}} | Speicheranzeige in Hex- und KOI-7-Format |
| Efilespec | Laden eines Programms zur Ausführung |
| Fs,f,b | Füllen eines Speicherblockes mit einer Byte-konstanten |
| FWs,fw | Füllen eines Speicherblockes mit einer Wort-konstanten |
| G{s}{,b1{,b2}} | Start der Programmausführung und Setzen von Unterbrechungspunkten |
| Hw1,w2 | Hexadezimale Summe und Differenz |
| Icommand tail | Eingabe einer Kommandozeile in den FCB (file control block) und Kommandopuffer |
| L{s{,f}} | Speicherausgabe im Mnemonikformat |
| Ms,f,d | Transport eines Speicherblockes |
| Rfilespec | Lesen einer Plattendatei in den Speicher |
| S{W}s | Ein- und Ausgabe von Speicherinhalten (Byte/Wort) |
| T{n} | Programmausführung im Testmodus |
| TS{n} | Testmodus und Anzeige aller Register |
| U{n} | Programmausführung im Direktmodus |
| US{n} | Direktmodus und Anzeige aller Register |
| V | Ausgabe der Speicherbelegung einer eingelesenen Datei |
| Wfilespec{,s,f} | Schreiben eines Blockinhaltes zur Platte |
| X{r} | Prüfen und Modifizieren ZVE-Register |

Bedeutung der in Tabelle 7 aufgeführten Parameter:

| | |
|---|---|
| b | Bytekonstante |
| b1 | Unterbrechungspunkt 1 |
| b2 | Unterbrechungspunkt 2 |
| d | Ziel für Daten |
| f | Endadresse (Offset) |
| n | Anzahl auszuführender Befehle |
| r | Register- oder Flagname |
| s | Startadresse, besteht aus {Segmentadresse:} Offset |
| s1 | zweite Startadresse |
| W | 16-bit-Wort |
| w | Wortkonstante |

Eine ausführliche Beschreibung des DDT86 und seiner Kommandos ist in der Anleitung für den Programmierer enthalten.

Wenn keine Dateispezifikation angegeben wurde, wird DDT86 ohne ein Testprogramm geladen. In diesem Fall dürfen die DDT86-Kommandos G, T und U nicht angewendet werden. Das Laden eines Testprogrammes erfolgt gewöhnlich mit dem E-Kommando.

Ist eine Dateispezifikation angegeben, wird DDT86 zusammen mit der spezifizierten Datei in den Speicher geladen. Es können die Kommandos G, T und U benutzt werden, um das Testprogramm unter

Steuerung des DDT86 abzuarbeiten. Wird in der Dateispezifikation der Dateityp weggelassen, wird der Typ CMD angenommen. DDT86 kann Testprogramme im Objektformat (H86) nicht direkt laden. Sie müssen erst in Kommandodateien (CMD) umgewandelt werden. Dazu dient das Dienstprogramm GENCMD.

Beispiele:

```
A>DDT86
```

Das Dienstprogramm DDT86 wird vom Laufwerk A in den Speicher geladen. DDT86 gibt als Meldung ein - (Minuszeichen) aus, wenn es eine Kommandoeingabe anfordert.

```
A>B:DDT86 TEST.CMD
```

DDT86 wird vom Laufwerk B geladen, danach das Testprogramm TEST.CMD vom Laufwerk A. DDT86 zeigt die Adresse an, ab der das Testprogramm geladen ist und gibt die Meldung - aus.

## 4.6. Kommando DIR (Verzeichnisausgabe)

Syntax:

```
DIR {d:}
DIR {filespec}
DIRS {d:}
DIRS {filespec}
```

Typ:

Residentes Kommando

Anwendung:

Die residenten DIR- und DIRS-Kommandos zeigen die Dateinamen an, die im Verzeichnis der angegebenen Platte katalogisiert sind. Das DIR-Kommando gibt die Dateinamen der laufenden Nutzernummer aus, die ein DIR-Attribut besitzen. DIR akzeptiert Dateigruppensymbole in der Dateispezifikation.

Das DIRS-Kommando zeigt die Dateinamen der laufenden Nutzernummer an, die ein SYS-Attribut besitzen. Obwohl von jedem Nutzerbereich auf einer Platte auf SYS-Dateien mit der Nutzernummer 0 zugegriffen werden kann, zeigt DIRS die SYS-Dateien aus dem Nutzerbereich 0 nur an, wenn als aktuelle Nutzernummer 0 angegeben ist. DIRS akzeptiert Dateigruppensymbole in der Dateispezifikation.

Werden weder ein Laufwerk noch eine Datei spezifiziert, zeigt DIR alle Dateien mit dem DIR-Attribut an, die sich auf der Platte im aktuellen Laufwerk befinden und die die aktuelle Nutzernummer haben. Analog zeigt DIRS die Dateien mit dem SYS-Attribut an. Wird nur das Laufwerkkennzeichen, aber keine Dateispezifikation angegeben, so zeigt DIR die Namen der DIR-Dateien mit der laufenden Nutzernummer an, die sich auf der spezifizierten Platte befinden, DIRS entsprechend die SYS-Dateien.

Wenn in der Dateispezifikation die Dateigruppensymbole enthalten sind, werden alle Dateien angezeigt, die den bzgl. der

Sonderzeichen festgelegten Regeln genügen. Stimmt kein Dateiname mit der Dateispezifikation überein oder sind keine Dateien im Verzeichnis katalogisiert, dann gibt DIR die Mitteilung aus: NO FILE.

Wenn auf der spezifizierten Platte nur Systemdateien (SYS) vorhanden sind, gibt DIR die Mitteilung aus: SYSTEM FILE(S) EXIST.

Existieren auf der spezifizierten Platte nur Dateien mit DIR-Attribut, dann gibt DIRS die Mitteilung aus: NON-SYSTEM FILE(S) EXIST. In der Laufwerkspezifikation dürfen keine Sonderzeichen verwendet werden.

Beispiele:

```
A>DIR
```

Alle DIR-Dateien, die auf der eingegliederten Platte im Laufwerk A unter der aktuellen Nutzernummer katalogisiert sind, werden auf dem Bildschirm angezeigt.

```
A>DIR B:
```

Alle DIR-Dateien der aktuellen Nutzernummer auf der Platte in B werden auf dem Bildschirm angezeigt.

```
A>DIR B:X.A86
```

Wenn die Datei X.A86 auf der Platte in B unter der aktuellen Nutzernummer exsistiert, zeigt DIR den Namen X.A86 auf dem Bildschirm an.

```
A>DIR *.BAS
```

Alle DIR-Dateien mit dem Typ BAS und der laufenden Nutzernummer auf der Platte im Laufwerk A werden angezeigt.

```
B>DIR A:X*.C?D
```

Alle DIR-Dateien mit der aktuellen Nutzernummer auf der Platte in A, deren Dateiname mit X beginnt und deren Dateityp mit C beginnt und mit D endet, werden angezeigt.

```
A>DIRS
```

Alle SYS-Dateien auf A mit der aktuellen Nutzernummer werden angezeigt.

```
A>DIRS *.CMD
```

Alle Dateien vom Typ CMD in A mit der aktuellen Nutzernummer und einem SYS-Attribut werden angezeigt.

## 4.7. DISKSET-Kommando

Syntax:

DISKSET

Anwendung:

Bei der Systemgenerierung, speziell bei der Erzeugung der Steuertabellen für den Externspeicher (Kommando GENDEF), werden die physischen (Spuranzahl, Sektorgröße usw.) und logischen (Anzahl der Systemspuren, Skew-Faktor, Verzeichnisgröße usw.) Parameter für den Zugriff zu den einzelnen Laufwerken festgelegt. Nach dem Systemstart sind genau diese Parameter gültig. Mit dem Kommando DISKSET können einige dieser Parameter auch nach dem Systemstart verändert werden. Danach hat der Nutzer die Möglichkeit, von dem SCP-Hausformat abweichende Disketten zu verarbeiten, ohne SCPX umgenerieren zu müssen.

Folgende Parameter sind einstellbar:

1) für 5,25"-Laufwerk (K5600.20)

   - 40 Spuren (48 tpi) oder 80 Spuren (96 tpi)
     Beachte: Im Laufwerk K5600.20 mit 48 tpi geschriebene Disketten sind im Laufwerk K5600.10 nicht lesbar!

   - Spurformatierung ab Spur 1:

     26 Sektoren a 128 Byte (skew = 6)
     16 Sektoren a 256 Byte
      8 Sektoren a 512 Byte
      9 Sektoren a 512 Byte
      4 Sektoren a 1024 Byte
      5 Sektoren a 1024 Byte

   - Blockgröße für Disketten mit

     Sektorgröße = 128 Byte : 1K Byte
     einfache Spurdichte (48 tpi) : 1K Byte oder 2K Byte
     sonst : 2K Byte

2) für 8"-Laufwerk, einfache Dichte (K5602.10)

   - Spurformatierung ab Spur 1:

     26 Sektoren a 128 Byte (skew = 6)
     15 Sektoren a 256 Byte
      8 Sektoren a 512 Byte
      4 Sektoren a 1024 Byte

   Blockgröße für Disketten mit

     Sektorgröße = 128 Byte : 1K Byte
     sonst : 2K Byte

3) für 8"-Laufwerk, doppelte Dichte (HF6400)

   - Verarbeitung von 8"-Disketten einfacher Dichte (siehe 2)
   - Spurformatierung ab Spur 1:

26 Sektoren a 256 Byte
15 Sektoren a 512 Byte
8 Sektoren a 1024 Byte

- Blockgröße ist immer 2K Byte

für alle Laufwerkstypen:

- Anzahl der Systemspuren von 1 bis 9 wählbar
- Anzahl der Verzeichniseinträge zwischen 64 und 128 wählbar

Erläuterungen zu den Laufwerksparametern enthält die Anleitung für den Systemprogrammierer.

DISKSET fordert zunächst die Eingabe eines Laufwerksbezeichners. Danach erhält der Nutzer Informationen über die aktuellen physischen Parameter des gewählten Laufwerks und kann, von DISKSET durch Menü-Angebote geführt, bestimmte Parameter verändern. Unzulässige Eingaben werden durch DISKSET ignoriert. Die Eingabe von CTRL/C führt zur Beendigung des Programms.

## 4.8. Kommando ED (Editor)

### Syntax:

ED input-filespec {d:|output-filespec}

### Typ:

Transientprogramm

### Anwendung:

Mit dem Dienstprogramm ED können Plattendateien erstellt und editiert werden. ED ist ein zeilenorientierter Texteditor. Das bedeutet, daß Textdateien Zeile für Zeile erstellt und geändert werden können. Auch auf einzelne Zeichen innerhalb einer Zeile kann Bezug genommen werden.

Mit ED kann nur die Datei erstellt oder verändert werden, die in der Dateispezifikation angegeben ist. ED benutzt einen Teil des Arbeitsspeichers als aktiven Textpuffer, in dem Zeichen hinzugefügt, geändert oder gelöscht werden. Mit dem A-Kommando wird die gesamte oder ein Teil der Datei in diesen Puffer gelesen. Umgekehrt dienen das W- oder das E-Kommando dazu, den gesamten Puffer oder einen Teil davon auf die Plattendatei zu schreiben.

Ein imaginärer Zeichenzeiger - CP genannt - steht am Beginn des Puffers, zwischen zwei Zeichen im Puffer oder an dessen Ende. Kommuniziert wird mit ED entweder im Kommandomodus oder im Eingabe-(Insert) Modus. Im Kommandomodus meldet sich ED mit einem *. Nach dieser Meldung können ED-Kommandos eingegeben werden, die z.B. Text aus dem Puffer lesen, den CP bewegen oder den ED-Modus ändern.

Tabelle 8: ED-Kommandos

| Kommando | Wirkung |
|---|---|
| nA | Übertragung von n Zeilen aus der Orginaldatei zum Textpuffer |
| 0A | Übertragung in den Textpuffer, bis dieser halb gefüllt ist |
| #A | Füllen des Textpuffers bis zum Ende (oder bis zum Ende der Datei) |
| B, -B | Bewegen des CP an den Beginn (B) oder das Ende (-B) des Textpuffers |
| nC, -nC | Bewegen des CP im Puffer um n Zeichen vorwärts (C) oder zurück (-C) |
| nD, -nD | Löschen von n Zeichen vor (-D) oder hinter (D) dem CP |
| E | Retten der neuen Datei und Rückgabe der Steuerung an das SCP |
| Fstring{^Z) | Suchen einer Zeichenkette (string) |
| H | Retten der neuen Datei, danach erneuter ED-Aufruf mit der neuen Datei als Orginaldatei |
| I | Wechsel zum Eingabemodus; mit CTRL/Z wird wieder zum Kommandomodus umgeschaltet |
| Istring{^Z} | Einfügen der Zeichenkette vor dem CP |
| JS1^ZS2^ZS3{^Z} | Kombiniertes Suchen, Einfügen und Streichen von Zeichenketten |
| nK, -nK | Löschen von n Zeilen ab CP |
| nL, -nL, 0L | Bewegen des CP um n Zeilen |
| nMcommands | n-maliges Ausführen des angegebenen Kommandos (bzw. ohne Angabe von n bis zum Ende der Datei) |
| n, -n | Bewegen des CP um n Zeilen und Anzeigen der Zeile |
| n: | Positionieren auf Zeile n |
| :ncommand | Ausführen des Kommandos bis zur Zeile n |
| Nstring{^Z} | Erweitertes Suchen einer Zeichenkette in der gesamten Datei |
| O | Rückkehr zur Originaldatei |
| nP, -nP | Bewegen des CP um n mal 23 Zeilen vorwärts (P) oder rückwärts (-P) und Anzeigen der folgenden 23 Zeilen auf dem Terminal |
| Q | Verlassen und Löschen der neuen Datei, Rückkehr zu SCP |
| R | Lesen der Datei X¤¤¤¤¤¤¤¤.LIB in den Puffer |
| Rfilespec{^Z} | Lesen der spezifizierten Datei in den Puffer |
| SD1^ZD2{^Z} | Zeichenkette ersetzen |
| nT, -nT, 0T | Anzeigen von n Zeilen |
| U, -U | Alle Kleinbuchstaben nach +U werden in Großbuchstaben umcodiert, die Umsetzung wird mit -U abgeschaltet |
| V, -V | Ein- und Ausschalten der Zeilennumerierung |
| 0V | Anzeige des freien Pufferraumes |
| nW | Schreiben von n Zeilen in die neue Datei |
| nX | Schreiben oder Anfügen von n Zeilen in bzw. an die Datei X¤¤¤¤¤¤¤¤.LIB |

Tabelle 8: (Fortsetzung)

| Kommando | Wirkung |
|---|---|
| nXfilespec{^Z} | Schreiben von n Zeilen in die spezifizierte Datei oder Anhängen von n Zeilen, falls sich das vorhergehende X-Kommando auf die gleiche Datei bezog |
| OX | Löschen der Datei Xxxxxxxx.LIB |
| OXfilespec{^Z} | Löschen der spezifizierten Datei |
| nZ | n Sekunden warten |

dabei bedeutet:

S1 search string
S2 insert string
S3 delete to string

D1 delete string
D2 insert string

Im Abschnitt 5. erfolgt eine ausführliche Beschreibung des Editors und seiner Kommandos.

Die Angabe der zweiten Dateispezifikation darf nur dann verwendet werden, wenn auch die erste Dateispezifikation bereits angegeben wurde und die Originaldatei nicht ersetzt werden soll. Die Dateispezifikation für die zweite Datei gibt die Datei an, die den geänderten Text aus der ersten Datei empfängt. Die erste Datei bleibt unverändert.

Wird für die zweite Datei nur das Laufwerk spezifiziert, dann erhält diese Datei automatisch Namen und Typ der ersten Datei.

Ist die der ersten Spezifikation entsprechende Datei nicht vorhanden, so wird sie neu eröffnet und ED teilt mit:

NEW FILE

In diesem Falle darf keine zweite Dateispezifikation angegeben werden. Wenn keine zweite Datei spezifiziert ist, dann wird die Originaldatei in den Typ BAK umbenannt und bleibt erhalten. Wenn ED eine Dateispezifikation mit dem Typ BAK übergeben wird, erstellt und rettet ED die neue editierte Version der BAK-Datei und löscht die Quelldatei. Wenn die Original-BAK-Datei gesichert werden soll, dann muß eine Umbenennung des Dateityps mit dem REN-Kommando vorausgehen, ED gibt der Originaldatei wieder den Typ BAK.

Wenn die zweite Dateispezifikation die gleiche wie die erste ist, dann erstellt und sichert ED die neu editierte Version der Ausgabedatei. Die originale Eingabedatei erhält den Dateityp BAK. Selbstverständlich dürfen nicht zwei Dateien mit den gleichen Namen im gleichen Nutzerbereich auf einer Platte stehen.

Wenn die Eingabedatei vorhanden ist, wird diese mit dem A-Kommando in Teilen oder vollständig in den Textpuffer

eingelesen.
Wenn die Dateigröße den Textpuffer nicht übersteigt, wird mit dem Kommando:

```
#a
```

die ganze Datei in den Textpuffer eingelesen.

Das i-(Eingabe)Kommando schaltet den ED auf Eingabemodus. In diesem Modus wird jedes eingegebene Zeichen sequentiell, mit der aktuellen CP-Position beginnend, in den Puffer eingetragen.

Die ED-Kommandos werden gewöhnlich als Buchstaben eingegeben. Die Kommandos, denen eine Zeichenfolge folgt, müssen mit CTRL/Z abschließen, wenn ihnen ein erneutes Kommando folgen soll.

Jedes Kommando, das als Großbuchstabe geschrieben ist, bewirkt, daß ED nachfolgende Zeichenketten bis zum CTRL/Z in Großbuchstaben umcodiert. Kommandos, geschrieben in Kleinbuchstaben, bewirken keine Umcodierung von Kleinbuchstaben in Zeichenketten.

Wenn Zeilennummern am linken Bildschirmrand erscheinen, dann haben sie die Form:

```
nnnnn:
```

wobei nnnnn eine ganze Zahl von 1 bis 24126 ist. Zeilennummern erscheinen nur bei der Anzeige. Sie werden weder im Puffer noch in der Datei geführt. Die Bildschirmzeile beginnt mit

```
:
```

wenn der CP am Anfang oder am Ende des Puffers steht.

Beispiele:

```
A>ED MYPROG.A86
```

Wenn die Datei nicht schon vorhanden ist, wird die Datei MYPROG.A86 auf der Platte in A erstellt. Die Meldung

```
:*
```

erscheint auf dem Bildschirm. Das bedeutet, daß der CP am Pufferanfang steht. Ist die Datei bereits vorhanden, dann werden mit

```
:*#a
```

der Puffer und danach mit dem Kommando

```
:*0p
```

der Bildschirm mit den ersten 23 Zeilen aus dem Puffer gefüllt.

Wird

```
:*e
```

eingegeben, beendet ED seine Arbeit. Die unveränderte Originaldatei erhält den Typ BAK und die geänderte Datei heißt

MYPROG.A86.

```
A>ED MYPROG.A86 B:NEWPROG.A86
```

Die Originaldatei MYPROG.A86 befindet sich in A. Diese Originaldatei verbleibt unverändert auf A, während die geänderte Datei mit der Spezifikation NEWPROG.A86 auf B gespeichert wird.

```
A>B:ED MYPROG.A86 B:
```

ED.CMD muß sich auf B befinden. Die Originaldatei ist MYPROG.A86 und befindet sich auf A. Sie bleibt unverändert, während die geänderte Datei MYPROG.A86 nach B gespeichert wird.

## 4.9. Kommando ERA (Löschen)

Syntax:

ERA filespec

Typ:

Residentes Kommando

Anwendung

Mit dem ERA-Kommando werden eine oder mehrere Dateien aus dem Verzeichnis einer Platte gelöscht. In der Dateispezifikation werden Dateigruppensymbole zugelassen. Verzeichnisplatz und Speicherraum der gestrichenen Dateien werden für spätere Benutzung und andere Dateien freigegeben.

Das ERA-Kommando muß mit Sorgfalt verwendet werden, da es alle angegebenen Dateien aus dem Plattenverzeichnis löscht. Das Kommando:

```
ERA{d:}*.*
```

fordert die Zustimmung des Bedieners an:

```
ALL (Y/N)?
```

Sollen alle Dateien gelöscht werden, ist mit Y zu antworten, anderenfalls mit N. Wenn keine Datei entsprechend der Spezifikation existiert, erscheint die Mitteilung

```
NO FILE
```

auf dem Bildschirm.

Beispiele:

```
A>ERA X.A86
```

Die Datei X.A86 wird aus dem Verzeichnis der Platte A gelöscht.

```
A>ERA *.PRN
```

Alle Dateien vom Typ PRN werden aus dem Verzeichnis in A gelöscht.

```
B>ERA A:MY*.*
```

Jede Datei auf A, deren Dateiname mit MY beginnt, werden aus dem Verzeichnis gelöscht.

```
A>ERA B:*.*
```

Alle Dateien auf B werden gelöscht. Um diese Operation ausführen zu können, muß auf die Frage

```
ALL (Y/N)?
```

mit Y geantwortet werden.

## 4.10. Kommando GENCMD (CMD-Generierung)

Syntax:

```
GENCMD filespec {8080 CODE[An,Bn,Mn,Xn] DATA[An,Bn,Mn,Xn]
       STACK[An,Bn,Mn,Xn] EXTRA[An,Bn,Mn,Xn] X1[...  }
```

Typ:

Transientprogramm

Anwendung:

Das Dienstprogramm GENCMD generiert aus der Ausgabedatei (H86) des ASM oder anderer Sprachprozessoren eine CMD-Datei. Der Dateispezifikation folgt eine wahlweise Parameterliste.

Das Schreiben von Programmen in Assemblersprache, die Transientkommandos werden sollen, setzt die Kenntnis von GENCMD voraus. Die Arbeit mit GENCMD ist im Detail in der Anleitung für den Systemprogrammierer beschrieben.

Die Parameterliste besteht aus bis zu neun Schlüsselworten mit einer zugehörigen Liste von Werten. Die Schlüsselworte sind:

```
8080 CODE DATA STACK EXTRA X1 X2 X3 X4
```

Das Schlüsselwort 8080 identifiziert die CMD-Datei für das 8080-Speichermodell, in dem sich Code und Daten befinden. Die restlichen Schlüsselworte definieren Segmentgruppen, die spezifische Speicherforderungen haben. Die Werte, die die Speicherforderungen definieren, werden durch Kommas getrennt, in eckige Klammern gesetzt und folgen jedem Schlüsselwort. Die Schlüsselworte mit den in eckigen Klammern stehenden zugehörigen Werten müssen vom nächsten Schlüsselwort durch ein Leerzeichen getrennt werden. Die Werte in den eckigen Klammern werden nachstehend definiert, wobei n eine hexadezimale Konstante aus ein bis vier Ziffern darstellt. Der Wert n repräsentiert einen Paragraphwert , wobei jeder Paragraph 16 Bytes lang ist. Der Paragraphwert korrespondiert zum Bytewert durch n * 16 oder hhhh0 hexadezimal.

| | |
|---|---|
| An | Laden der Gruppe ab der absoluten Adresse n |
| Bn | Anfang der Gruppe bei Adresse n in der Hexadezimaldatei |
| Mn | Die Gruppe fordert ein Minimum von n * 16 Bytes. |
| Xn | Die Gruppe kann bis zu n * 16 Bytes adressieren. |

Das 8080-Schlüsselwort wird für Programme benutzt, die vom 8-bit-Prozessor zum 16-bit-Prozessor konvertiert wurden. Die Programme werden in einen Bereich geladen, in dem sich Daten und Code befinden. Das Codesegment im Programm muß ab Adresse 100H beginnen.
Für jede Gruppe, die ab einer absoluten Speicheradresse geladen werden muß, ist An zu benutzen. Der A-Wert darf nicht benutzt werden, wenn nicht bekannt ist, daß der geforderte Absolutbereich während der Programmabarbeitung verfügbar ist.

Bn wird benutzt, wenn die Hex-Datei keine Informationen enthält, die die Segmentgruppen identifizieren. Dieser Wert ist nicht notwendig, wenn die H86-Datei die Ausgabedatei des ASM86-Assemblers ist.
Wenn ein Datensegment vorliegt, das einen nichtinitialisierten Datenbereich am Segmentende besitzt, wird Mn verwendet.
Xn dient dazu, dem Programm einen größeren Datenbereich als das durch Mn angegebene Minimum zuzuweisen, wenn er verfügbar ist.

Beispiele:

```
A>GENCMD MYFILE
```

Die Datei MYFILE.H86 wird von A gelesen. Die Ausgabedatei MYFILE.CMD wird nach A zurückgeschrieben. Die Eingabedatei (H86) enthält Informationen, die das Programm für die Arbeit mit einem bestimmten Speichermodell kennzeichnen.

```
B>GENCMD MYFILE CODE[A40] DATA[M30,XFFF]
```

Die Datei MYFILE.H86 wird von B gelesen und MYFILE.CMD nach B geschrieben. Die Codegruppe muß ab 400H geladen werden. Die Datengruppe fordert ein Minimum von 300H Bytes. Wenn es möglich ist, kann das Programm bis FFF0H Bytes nutzen.

## 4.11. Kommando HELP (Bedienerhilfe)

Syntax:

```
HELP {topic}{subtopic1 subtopic2 ... subtopic8}{[P]}
```

Typ:

Transientprogramm

Anwendung:

Das HELP-Kommando informiert in zusammengefaßter Weise über alle in der vorliegenden Anleitung für den Bediener beschriebenen Kommandos. Um auch gezielt spezielle Informationen zu erhalten, können dem Kommandoschlüsselwort als Parameter Suchbegriffe (topic) und diesem untergeordnete Suchbegriffe (subtopic) zugeordnet werden. Die Suchbegriffe sind z. B. die Kommandoschlüsselworte, untergeordnete Suchbegriffe sind z. B. Kommandoparameter usw.
Wird HELP ohne Parameter eingegeben, so wird die Liste aller möglichen Suchbegriffe auf dem Bildschirm angezeigt. Wird HELP mit einem Kommandoschlüsselwort als Parameter eingegeben, so wird die Beschreibung dieses Kommandos entsprechend der vorliegenden Schrift angezeigt, einschließlich der Liste der nachgeordneten

Suchbegriffe. HELP mit einem Suchbegriff und einem nachgeordneten Suchbegriff bewirkt das Anzeigen der gewünschten Information über den nachgeordneten Suchbegriff. Die Reihenfolge der nachgeordneten Suchbegriffe ist einzuhalten.

Wenn mit HELP die gewünschten Informationen angezeigt wurden, erscheint zum Abschluß die Meldung HELP> auf dem Bildschirm. Danach können entweder weitere Informationen zu anderen Suchbegriffen angefordert oder durch Drücken der RETURN-Taste zum SCP zurückgekehrt werden. Die Suchbegriffe bzw. nachgeordneten Suchbegriffe können abgekürzt werden, gewöhnlich genügen ein oder zwei Buchstaben. HELP mit der wahlweisen Angabe [P] bewirkt, daß die Bildschirmanzeige ohne Pause in Blöcken zu 23 Zeilen erfolgt. HELP ohne [P] bewirkt die Anzeige einer Bildschirmseite und die Ausschrift

```
Press enter to continue
```

die mit einer Eingabe zu quittieren ist.

Beispiele:

```
A>HELP
```

Das Kommando listet alle Suchbegriffe aus, für die Informationen abgefragt werden können.

```
A>HELP STAT
```

Dieses Kommando bewirkt die Anzeige allgemeiner Informationen über das STAT-Kommando. Es zeigt auch alle möglichen nachgeordneten Suchberiffe an.

```
A>HELP STAT OPTIONS
```

Dieses Kommando enthält den nachgeordneten Suchbegriff options. Im Ergebnis zeigt HELP die mit STAT zusammenhängenden wahlweisen Angaben an.

```
A>HELP ED
```

Dieses Kommando gibt allgemeine Informationen über den Editor ED aus.

```
A>HELP ED COMMANDS
```

In diesem Fall werden die internen Kommandos des ED angezeigt.

## 4.12. INIT-Kommando (Formatieren von Disketten)

Syntax:

INIT

Typ:

Transientes Kommando

Anwendung:

INIT dient zur Formatierung einer Platte, d.h. zur Festlegung der physischen Struktur. Für Platten, die unter dem Betriebssystem SCP 1700 benutzbar sein sollen, werden zwei weitere Funktionen ausgeführt:

- Initialisierung des SCP-1700-Verzeichnisbereichs auf der Platte
- Belegen defekter Datenspuren mit einer Sperrdatei

Nach dem Aufruf

```
INIT
```

meldet sich das Initialisierungsprogramm mit

```
INIT Vx.y

Disk drive?
```

und erwartet die Eingabe einer Laufwerksbezeichnung (A-P).
Alle Eingaben sind mit (CR) abzuschließen. Danach fragt INIT, ob die zu formatierende Platte eine SCP-1700-Platte sein soll.

```
INIT SCP1700 disk using current drive parameters (N/<CR>)?
```

Ist die Antwort (CR), entnimmt INIT die für die Formatierung notwendigen Plattenparameter den entsprechenden Gerätesteuertabellen.
Bei Eingabe von N erfragt INIT nacheinander die benötigten Parameter

```
Tracks /Disk     :      (Spurenanzahl)
Sectors/Tracks   :      (Sektoren/Spur)
Bytes  /Sector   :      (Sektorgröße)
Track 0 handling?       (Behandlung von Spur 0)
```

Die Spuranzahl wird vom System selbst festgelegt (80 Spuren bei 5,25", 77 Spuren bei 8"). Die folgenden beiden Werte sind durch den Bediener einzugeben. Ist die Spur 0 abweichend von den übrigen Spuren zu formatieren (einfache Dichte, Sektorgröße 128 Bytes, 26 Sektoren), dann ist die Frage 'Track 0 handling?' mit Y (Yes), sonst mit N (No) zu beantworten.

Vor Beginn des Formatierens wird durch INIT abgefragt, in welcher Reihenfolge die Sektoren physisch innerhalb einer Spur angeordnet werden sollen (interleave).
Standard ist interleave = 1 (Eingabe <CR>).

Der Formatiervorgang beginnt nach entsprechender Reaktion des Bedieners auf die Anforderung:

```
Insert scratch disk in drive x

Type <CR> to continue
```

(Platte in Laufwerk x einlegen und mit (CR) fortsetzen.)

Nach Abschluß des Formatierens fragt INIT:

Init more disk (Y/N)?

Bei Ausgabe von Y (Yes) kann auf dem gleichen Laufwerk die nächste Platte mit den gleichen Parametern initialisiert werden.

Bei Angabe von N (No) fragt INIT:

Init another disk (Y/N)?

Nach Eingabe von Y (Yes) kann die Initialisierung einer Platte mit anderen Parametern vorbereitet werden, sonst wird INIT beendet.

Fehler:

Eine falsche Laufwerksangabe führt zur Fehlermeldung:

Illegal diskette drive

Danach wird mit einer erneuten Eingabeanforderung fortgesetzt. Fehler beim Formatieren einer Spur werden durch

Disk error: Retry(R)?

Disk ERROR: Retry(R) or Ignore(I) or Cancel(C)

angezeigt.

Die Eingabe von R bewirkt die Wiederholung des Plattenzugriffs, während C zum Abbruch des Programms INIT führt. Jede andere Eingabe führt zum Abbruch des Formatierens, falls die Frage

Init SCP1700 disk using current drive parameters (N/<CR>)?

mit N beantwortet wurde.

Beim Initialisieren von SCP-Platten wird beim Auftreten von nicht durch Wiederholung behebbaren Zugriffsfehlern wie folgt verfahren:

(1) Der Fehler trat beim Formatieren einer Systemspur auf.
Es wird mit der Formatierung der nächsten Spur fortgesetzt.
Nach Abschluß des Initialisierens erscheint die Ausschrift:

Disk not usable as SCP system disk

(Die Platte ist nicht als Systemplatte, d.h. zum Systemstart benutzbar.)

(2) Der Fehler trat beim Formatieren der Spur auf, die das Dateiverzeichnis enthält.
Die Formatierung wird mit der Ausschrift

Disk not usable

abgebrochen. Die Platte ist unter SCP 1700 nicht nutzbar.

(3) Der Fehler trat beim Formatieren einer Datenspur auf.
Diejenigen logischen Datenblöcke, die mit der fehlerhaften Spur korrespondieren, werden im Dateiverzeichnis als belegt

gekennzeichnet. Danach wird mit der Formatierung der nächsten Spur fortgesetzt. Nach Abschluß des Initialisierens erscheint die Ausschrift:

Disk contains bad data tracks

(Die Platte enthält fehlerhafte Datenspuren.)

## 4.13. Kommando PIP (Dateikopie)

Syntax:

PIP dest-file{[Gn]}|dev=src-file{[options]}|dev{[options]}

Typ:

Transientprogramm

Anwendung:

PIP kopiert eine oder mehrere Dateien von einer Platte und/oder Nutzernummer zu einer anderen. Eine Datei kann nach dem Kopieren umbenannt und zwei oder mehrere Dateien können zu einer verbunden werden. PIP kann auch eine Textdatei von der Platte auf den Drucker oder ein anderes logisches Hilfsgerät für die Ausgabe kopieren, eine Plattendatei durch Eingabe von dem Terminal oder einem anderen logischen Eingabegerät erstellen und Daten von einem logischen Eingabegerät zu einem logischen Ausgabegerät übertragen. Deshalb trägt das Programm die englische Bezeichnung Periphal Interchange Program.

### 4.13.1. Kopie einer Datei

Syntax:

```
PIP d:{[Gn]}=source-filespec{[options]}
PIP dest-filespec{[Gn]}=d:{[options]}
PIP dest-filespec{[Gn]}=source-filespec{[options]}
```

Anwendung:

Die erste Form zeigt den einfachsten Weg, um eine Datei zu kopieren. PIP sucht nach der durch source-filespec angegebenen Datei auf dem Standardlaufwerk oder dem wahlweise angegebenen Laufwerk. PIP kopiert diese Datei zum spezifizierten Laufwerk d: und gibt der Kopie den gleichen Namen wie source-filespec. [Gn] kann angegeben werden, um die Kopie unter der Nutzernummer n zu erstellen. [Gn] ist die einzig erlaubte Parameterangabe für die Zieldatei. Für die Quelldateispezifikation (source-filespec) können mehrere wahlweise Angaben kombiniert verwendet werden. Näheres hierzu ist im Abschnitt 4.13.6. zu finden.

Die zweite Form ist eine Variation der ersten. PIP sucht die spezifizierte Datei dest-filespec auf d:, kopiert sie auf das festgelegte Laufwerk und gibt ihr den gleichen Namen wie dest-filespec.

Die dritte Form zeigt, wie eine Datei nach dem Kopieren umbenannt werden kann. Sie kann zur gleichen Platte und Nutzernummer bzw. zu einer anderen Platte und/oder Nutzernummer kopiert werden.

Die Regeln für die Dateispezifikationen sind die gleichen. PIP sucht die spezifizierte Quelldatei (source-filespec), kopiert sie an die vorgegebene Stelle und gibt ihr den in dest-filespec angegebenen Namen. Zu beachten ist, daß PIP stets nur von und zur aktuellen Nutzernummer arbeitet, wenn nicht mit [Gn] etwas anderes vereinbart wird.

Vor dem Start von PIP ist es notwendig, sich zu vergewissern, daß auf der Zielplatte ausreichend freier Raum zum Kopieren einer oder mehrerer Dateien vorhanden ist. Wird eine Datei durch eine neue ersetzt, benötigt PIP genügend Raum für die Kopie, da erst nach dem Kopieren die alte Datei gelöscht wird (siehe Abschnitt 4.15.2.).

Daten werden zuerst in eine temporäre Datei kopiert, um abzusichern, daß die gesamte Datei innerhalb des verfügbaren Raumes auf der Platte aufgebaut werden kann. PIP gibt dieser temporären Datei den Dateinamen, der für die Zieldatei (Kopie) spezifiziert ist und weist dieser den Typ ¤¤¤ zu. War der Kopiervorgang erfolgreich, dann ändert PIP den Typ der temporären Datei in den für die Kopie vorgesehen um.

Wird ein Kopiervorgang ausgeführt und die angegebene Zieldatei existiert bereits mit dem gleichen Namen, so wird diese vor der Umbenennung der temporären Datei gelöscht. Die Dateiattribute (SYS, DIR, RW, RO) werden mit den Dateien übertragen.

Wenn eine existierende Datei das Attribut Read-Only(RO) besitzt, fragt PIP, ob sie gelöscht werden soll. Die Frage ist mit Y oder N zu beantworten. Zum Überspielen von Read-Only-Dateien wird der Parameter W genutzt.

Bei PIP können die wahlweise angebbaren Parameter jedem Quellnamen folgen, aber es gibt nur eine zulässige Angabe für die Zieldateispezifikation ([Gn] - Kopie zur Nutzernummer n). Wahlweise Angaben werden in eckige Klammern gesetzt. Bei der Quelldatei können mehrere wahlweise Angaben stehen, die unmittelbar aufeinander folgen oder durch Leerzeichen getrennt werden.

Wahlweise Parameterangaben können PIP veranlassen, den Kopiervorgang zu prüfen, Dateien mit dem Attribut SYS zu lesen, Read-Only-Dateien zu überschreiben, in oder aus einem Bereich mit einer spezifizierten Nutzernummer zu kopieren, Kleinbuchstaben in Großbuchstaben umzuwandeln und vieles mehr.

Beispiele:

```
A>PIP B:=A:oldfile.dat

A>PIP B:oldfile.dat=A:
```

Beide Kommandos haben die gleiche Wirkung. Die Datei oldfile.dat wird von A gelesen und nach B kopiert. Diese Form wird als Kurzform des PIP-Kommandos bezeichnet, weil der Quell- oder Zielname nur ein Laufwerk ist und keinen Dateinamen enthält. Damit ist es nicht möglich Dateien von einem Laufwerk und einer Nutzernummer zum gleichen Laufwerk und der gleichen Nutzernummer zu kopieren. Für die Zieldatei ist ein anderes Laufwerk oder eine andere Nutzernummer anzugeben (siehe Abschnitt 4.13.6. und 4.18.).

Die zweite Kurzform liefert das gleiche Ergebnis wie die erste. PIP sucht einfach nach der Datei oldfile.dat auf A, dem Laufwerk der Quelldatei.

```
A>PIP B:newfile.dat=A:oldfile.dat
```

Dieses Kommando kopiert die Datei oldfile.dat von A nach B und nennt diese Datei auf B newfile.dat. Die Datei oldfile.dat verbleibt auf A. Dies ist die lange Form des PIP-Kommandos, weil es auf beiden Seiten Dateinamen enthält.

```
A>PIP newfile.dat=oldfile.dat
```

Mit diesem Kommando wird eine Datei von einem Laufwerk und einer Nutzernummer (gewöhnlich Nummer 0, weil im SCP Nutzernummer 0 Standard ist) zu dem gleichen Laufwerk und der gleichen Nummer kopiert. Das ergibt effektiv zweimal die gleiche Datei mit verschiedenen Namen auf einem Laufwerk unter der gleichen Nutzernummer.

```
A>PIP B:PROGRAM.BAK=A:PROGRAM.DAT[G1]
```

Dieses Kommando kopiert die Datei PROGRAM.DAT vom Nutzer 1 auf Laufwerk A zur aktuellen Nutzernummer auf Laufwerk B und nennt die Datei auf B in den Typ BAK um.

```
B>PIP program2.dat=A:program1.dat[E V G3]
```

PIP kopiert die Datei program1.dat von A und echot [E] die Übertragung zum Terminal, vergleicht [V], ob die beiden Kopien exakt übereinstimmen und nimmt die Datei program1.dat vom Nutzer 3 auf A. Da für das Ziel kein Laufwerk spezifiziert ist, kopiert PIP die Datei automatisch zur Standardnutzernummer und zum Standardlaufwerk, in diesem Fall B.

### 4.13.2. Kopieren mehrerer Dateien

Syntax:

```
PIP d:{[Gn]}={d:}wildcard-filespec{[options]}
```

Anwendung:

Wird ein Dateigruppensymbol in der Quellspezifikation verwendet, dann kopiert PIP die entsprechenden Dateien nacheinander zur Zielplatte, wobei für jede Datei der Originalname erhalten bleibt. PIP gibt die Mitteilung COPYING und die Namen aller kopierten Dateien aus. Sind Zielplatte und Nutzernummer gleich den in der Quelle spezifizierten, gibt PIP eine Fehlermitteilung aus und bricht den Kopiervorgang ab.

Beispiele:

```
A>PIP B:=A:*.CMD
```

PIP kopiert alle Dateien mit dem Typ CMD von A nach B.

```
A>PIP B:=A:*.*
```

PIP kopiert alle Dateien von A nach B. Dieses Kommando wird genutzt, um eine Kopie zur Sicherung der Platte herzustellen. Dabei ist aber zu beachten, daß das SCP-System, das die Systemspuren belegt, nicht kopiert wird. COPYDISK kopiert auch das System.

```
A>PIP B:=A:PROG????.*
```

PIP kopiert alle Dateien, die mit PROG beginnen von A nach B, unabhängig vom Dateityp.

```
A>PIP B:[G1]=A:*.A86
```

PIP kopiert alle Dateien mit dem Dateityp A86 von der Platte in A mit der aktuellen Nutzernummer nach B in den Bereich mit der Nutzernummer 1. Die aktuelle Nutzernummer ist 0, wenn sie nicht mit dem USER-Kommando geändert wurde.

Die Kommandos DIR, TYPE, ERA und weitere Kommandos greifen nur auf Dateien zu, die sich im gleichen Nutzerbereich befinden, von dem sie gestartet wurden (siehe Abschnitt 4.18.).

### 4.13.3. Fügen von Dateien

Syntax:

```
PIP dest-file{[Gn]}=src-file{[opt]},file{[opt]}{,file{[opt]}...}
```

Anwendung:

Diese Kommandoform ermöglicht es, zwei oder mehr Quelldateien zu spezifizieren. PIP kopiert diese als Quelle spezifizierten Dateien von links nach rechts und vereinigt sie zu einer Datei, deren Name durch die Spezifikation der Zieldatei gegeben ist. Soll die Zieldatei eine spezifizierte Nutzernummer n erhalten, ist [Gn] bei der Zieldatei anzugeben. Für jede Quelldatei können ein oder mehrere wahlweise Parameter festgelegt werden.

Die meisten wahlweisen Parameter veranlassen PIP, Dateien zeichenweise zu kopieren. In diesen Fällen testet PIP auf das Dateiendezeichen CTRL/Z. Außer den folgenden, bewirken alle wahlweisen PIP-Parameter eine Zeichenübertragung:

Gn, K, O, R, V und W.

Das Kopieren von oder zu logischen Geräten erzwingt ebenfalls eine Zeichenübertragung.

Während der Zeichenübertragung kann durch Eingabe eines beliebigen Zeichens über die Tastatur das Zusammenfügen von Dateien abgebrochen werden. PIP sucht nach dem Dateiende (CTRL/Z). Wird eine Zeichenübertragung ausgeführt, dann wird diese beim Erkennen eines CTRL/Z abgebrochen.
Zum Zusammenfügen von Maschinencodedateien wird der Parameter [O] benutzt. Dieser veranlaßt PIP, das Dateiendezeichen CTRL/Z zu ignorieren.

Beispiele:

```
A>PIP NEWFILE=FILE1,FILE2,FILE3
```

Die drei Dateien FILE1, FILE2, FILE3 werden von links nach rechts verbunden und in die Datei NEWFILE.¤¤¤ kopiert. Wenn der Kopiervorgang erfolgreich beendet wurde, wird die Datei NEWFILE.¤¤¤ in NEWFILE umbenannt. Alle Quell- und Zieldateien befinden sich auf der Platte im Standardlaufwerk A.

```
A>PIP B:X.A86=Y.A86,B:Z.A86
```

Die Datei Y.A86 auf A wird mit Z.A86 von B verbunden und in die temporäre Datei X.¤¤¤ auf B kopiert. Die Datei X.¤¤¤ wird in die Datei X.A86 umbenannt, wenn PIP den Kopiervorgang erfolgreich abgeschlossen hat.

## 4.13.4. Kopieren von Dateien zu und von Hilfsgeräten

Syntax:

```
PIP dest-filespec{[Gn]}=source-filespec{[options]}
      AXO:                AXI:{[options]}
      CON:                CON:{[options]}
      PRN:                NUL:
      LST:                EOF:
```

Anwendung:

Diese Form ist ein Spezialfall des PIP-Kommandos, mit dem eine Datei von einer Platte zu einem Gerät, von einem Gerät zu einer Platte oder von einem Gerät zu einem anderen kopiert werden kann. Die Dateien müssen druckbare Zeichen enthalten. Jedes Peripheriegerät kann einem logischen Namen zugewiesen werden, der es als ein Quellgerät, das Daten senden kann, oder als ein Zielgerät, das Daten empfangen kann, definiert. Jedem logischen Gerätenamen muß ein Doppelpunkt (:) folgen, damit er nicht mit einem Dateinamen verwechselt wird.

Durch Drücken einer beliebigen Taste kann ein Kopiervorgang, in dem ein logisches Gerät als Quelle oder Ziel verwendet wird, abgebrochen werden.

Logische Gerätenamen sind:

CON: Terminal: physisches Gerät, das CON: zugewiesen ist Wenn es als Quelle genutzt wird, ist es gewöhnlich die Tastatur; wenn es als Ziel genutzt wird, ist es gewöhnlich der Bildschirm.
AXI: Hilfsein- oder Ausgabegerät
AXO: Hilfsausgabegerät
LST: Zielgerät, das LST: zugewiesen ist, im allgemeinen der Drucker.

Es gibt drei Gerätenamen mit besonderer Bedeutung:

NUL: Quellgerät, das 40 hexadezimale Nullen produziert.
EOF: Quellgerät, das ein Zeichen CTRL/Z produziert (Dateiendezeichen).

PRN: Druckgerät mit Tabulatoren aller 8 Spalten, Zeilennummern und Seitenvorschub aller 60 Zeilen.

Beispiele:

```
B>PIP PRN:=CON:, MYDATA.DAT
```

Zuerst werden Zeichen vom Terminal (im allgemeinen der Tastatur) gelesen und direkt zum Druckgerät gesendet. Mit der Eingabe eines CTRL/Z wird die Tastatureingabe beendet. Jetzt setzt PIP mit dem Lesen der Datei MYDATA.DAT von B fort. Da PRN: das Zielgerät ist, werden Tabulatoren wirksam, Zeilennummern hinzugefügt und aller 60 Zeilen eine neue Seite begonnen.

```
A>PIP B:FUNFILE.SUE=CON:
```

Wenn CRT: auf CON: zugewiesen ist, dann wird alles, was über das Terminal eingegeben wird, zur Datei FUNFILE.SUE auf Laufwerk B übertragen. Die Tastatureingabe wird mit CTRL/Z beendet.

```
A>PIP LST:=CON:
```

Wenn CRT: auf CON: zugewiesen ist, werden alle Terminaleingaben zum Listengerät, im allgemeinen dem Drucker übertragen. Die Eingabe wird mit CTRL/Z beendet.

```
A>PIP LST:=B:DRAFT.TXT[T8]
```

Die Datei DRAFT.TXT wird zum Druckgerät übertragen. Alle Tabulatoren werden auf ein Vielfaches von 8 positioniert.

```
A>PIP PRN:=B:DRAFT.TXT
```

PIP veranlaßt die Ausgabe der Datei DRAFT.TXT auf dem Listengerät. Tabulatoren sind automatisch wirksam, Zeilennummern werden hinzugefügt und aller 60 Zeilen beginnt eine neue Seite.

### 4.13.5. Modus für Mehrfachkommandos

Syntax:

PIP

Anwendung:

Diese Kommandoform startet das PIP. Danach können mehrere Kommandozeilen eingegeben werden, während PIP im Speicher verbleibt.

PIP gibt als Kommandoanforderung * auf dem Bildschirm aus. Danach können beliebige, dem PIP-Format entsprechende Kommandos eingegeben werden. Der Modus für Mehrfachkommandos wird durch Eingabe eines RETURN nach dem * verlassen. Eine leere Kommandozeile veranlaßt das PIP, seine Arbeit abzubrechen und das SCP zu aktivieren.

Beispiele:

```
A>PIP

*NEWFILE=FILE1,FILE2,FILE3

*APROG.CMD=BPROG.CMD

*A:=B:X.A86

*B:=*.*

*
```

Das erste Kommando startet PIP. Die Kommandofolge nach dem * entspricht den bereits erläuterten Beispielen, bei denen die Parameter an die PIP-Kommandos angehängt wurden. PIP wird hier nicht nach jedem Kommando neu in den Speicher geladen.

### 4.13.6. Wahlweise Parameter des PIP

Anwendung:

Wahlweise Parameter steuern die Bearbeitung der Quelldateien in besonderer Art. Es ist möglich, mit Tabulatoren zu arbeiten, Großbuchstaben in Kleinbuchstaben umzuwandeln, Textteile herauszulösen, die Kopie mit dem Orginal zu vergleichen und vieles mehr.

Die wahlweisen Parameter sind in Tabelle 9 ausgelistet. Für n ist eine Zahl einzusetzen und s stellt eine Zeichenfolge dar, die mit CTRL/Z beendet wird. Die wahlweisen Angaben sind in eckige Klammern [] einzuschließen. Bei den Parametern , die einen numerischen Wert fordern, dürfen zwischen Buchstaben und Zahl keine Leerzeichen stehen. Hinter einer Zieldatei kann der [Gn]-Parameter angegeben werden. Eine Parameterliste ist eine Folge von einzelnen Buchstaben und numerischen Werten, die wahlfrei durch Leerzeichen voneinander getrennt und in eckige Klammern eingeschlossen werden müssen.

Tabelle 9: Wahlweise Parameter des PIP

| Parameter | Funktion |
|---|---|
| Dn | Löschen aller Zeichen nach Spalte n. Dieser Parameter folgt einer Quelldatei, die eine für das Zielgerät zu lange Zeile enthält, z.B. ein 80-Zeichendrucker. Die Zahl n sollte die maximale Spaltenzahl des Zielgerätes sein. |
| E | Übertragung des Echos zur Bedieneinheit. Folgt dieser Parameter einem Quellnamen, zeigt PIP die Daten der Quelldatei auf der Bedieneinheit an, während der Kopiervorgang abläuft. Die Quelldatei muß druckbare Zeichen enthalten. |
| F | Herausfiltern von Seitenvorschubzeichen(FF). Wenn dieser Parameter einem Quellnamen folgt, löscht PIP alle in den Quelldaten enthaltenen FF-Zeichen. Soll die Seitenlänge der Zieldatei gegenüber der Quelldatei geändert werden, werden der F-Parameter zum Löschen der alten FF und der P-Parameter zum Einfügen neuer FF in die Zieldatei benutzt. |
| Gn | Übernehmen der Quelldatei vom oder Übertragen nach dem Nutzerbereich n. Folgt dieser Paramater einer Quelldatei, sucht PIP im Verzeichnis der Nutzernummer n nach der Quelldatei. Wenn er aber einer Zieldatei folgt, überträgt PIP die Zieldatei in den spezifizierten Nutzerbereich n. Die Nutzernummer liegt im Bereich von 0 bis 15. |
| H | Übertragung von Hex-Daten. PIP prüft alle Daten, ob sie den Forderungen des hexadezimalen Dateiformats entsprechen. Wenn Fehler erkannt werden, erfolgen Mitteilungen auf der Bedieneinheit. |
| I | Ignorieren von :00-Sätzen bei der Übertragung von Hex-Dateien. Der I-Parameter setzt automatisch den H-Parameter. |
| K | Beim Kopieren von mehreren Dateien wird die Ausschrift des jeweiligen Dateinamens auf dem Bildschirm unterdrückt. |
| L | Konvertieren von Großbuchstaben der Quelldatei in Kleinbuchstaben der Zieldatei. Dieser Parameter folgt dem Quellgerät oder dem Dateinamen. |

Tabelle 9: (Fortsetzung)

| Parameter | Funktion |
|---|---|
| N | Hinzufügen von Zeilennummern in der Zieldatei. Wenn dieser Parameter dem Quelldateinamen folgt, fügt PIP jeder kopierten Zeile eine Zeilennummmer hinzu, die mit 1 beginnt und in Stufen von 1 erhöht wird. Der Zeilennummer folgt ein Doppelpunkt. Wenn N2 spezifiziert wird, fügt PIP der Zeilennummer führende Nullen hinzu und nach der Nummer einen Tabulator. Falls der T-Parameter auch gesetzt ist, tritt die Tabulatorwirkung ein. |
| O | Übertragung von Objektdateien für spätere Maschinencodeerzeugung (keine druckbaren Zeichen). PIP überträgt keine Dateiendezeichen CTRL/Z. Dieser Parameter wird zum Fügen von Objektcodedateien benutzt. |
| Pn | Einstellen der Seitenlänge. n gibt die Zeilenzahl pro Seite an. Wenn dieser Parameter die Quelldatei modifiziert, fügt PIP zu Beginn der Zieldatei einen Seitenvorschub ein und danach immer nach n Zeilen. Ist n = 1 oder nicht spezifiziert, dann beträgt die Seitenlänge 60 Zeilen. Wird die F-Option auch angegeben, ignoriert PIP die Seitenvorschub-Zeichen (FF) in den Quelldateien und fügt neue FF in die Zieldateien entsprechend der mit n definierten Seitenlänge ein. |
| Qs | Beenden des Kopiervorganges vom Quellgerät beim Erkennen der Zeichenkette s. Wird zusätzlich ein S-Parameter angewendet, kann damit ein Teil der Quelldatei ausgewählt werden. Die Zeichenkette s muß mit CTRL/Z enden. |
| R | Lesen von Systemdateien (SYS). Normalerweise ignoriert PIP Dateien, die mit SYS-Attribut im Plattenverzeichnis gekennzeichnet sind. Wenn aber dieser Parameter einem Quelldateinamen folgt, dann kopiert PIP Systemdateien einschließlich ihrer Attribute zum Ziel. |
| Ss | Beginn des Kopierens der Quelldatei ab Zeichenkette s. Die Zeichenkette muß mit CTRL/Z enden. Wird dieser Parameter zusammen mit dem Q-Parameter angewendet, wird ein Teil der Quelldatei kopiert. Sowohl Anfangszeichenkette als auch Endezeichenkette werden in die Zieldatei kopiert. |
| Tn | Tabulatorerweiterung. Wenn dieser Parameter einem Quelldateinamen folgt, modifiziert PIP jeden Tabulator (CTRL/I). PIP erweitert jedes CTRL/I um soviel Leerzeichen, daß das nächste Zeichen in eine durch n teilbare Spalte gelangt. |

Tabelle 9: (Fortsetzung)

| Parameter | Funktion |
|---|---|
| U | Konvertieren von Kleinbuchstaben in der Quelldatei in Großbuchstaben in der Zieldatei. Dieser Parameter ist nach dem Quellgerät oder Quelldateinamen anzugeben. |
| V | Überprüfen der Kopie auf Korrektheit. PIP vergleicht Ziel- und Quelldatei, um zu sichern, daß der Kopiervorgang korrekt erfolgte. Das Ziel muß eine Plattendatei sein. |
| W | Überschreiben von Dateien mit RO Read-Only-Attribut. Normalerweise fragt PIP über das Terminal an, ob eine existierende Zieldatei mit RO-Attribut überschrieben werden soll. Folgt dieser Parameter dem Quelldateinamen, entfällt die Abfrage über die Bedieneinheit und PIP überschreibt die existierende RO-Datei. Wenn mehrere Quelldateien im Kommando angegeben werden, darf dieser Parameter nur der letzten Datei in der Folge angehängt werden. |
| Z | Nullsetzen des Paritätsbits. Wenn dieser Parameter dem Quellnamen folgt, setzt PIP das Paritätsbit eines jeden Datenbytes in der Zieldatei auf Null. Die Quelle muß Zeichen als Daten enthalten. |

Beispiele:

```
A>PIP NEWPROG.A86=CODE.A86[L],DATA.A86[U]
```

Dieses Kommando bildet die Datei NEWPROG.A86 auf A durch Verbinden der zwei Dateien CODE.A86 und DATA.A86, die sich auf A befinden. Während des Kopiervorganges wird die Datei CODE.A86 in Kleinbuchstaben und die Datei DATA.A86 in Großbuchstaben umgesetzt.

```
A>PIP CON:=WIDEFILE.A86[D80]
```

Dieses kommando schreibt die Zeichendatei WIDEFILE.A86 vom Laufwerk A zum Bildschirm und löscht dabei alle Zeichen, die nach der 80. Spalte folgen.

```
A>PIP B:=LETTER.TXT[E]
```

Die Datei LETTER.TXT auf A wird als Datei LETTER.TXT nach B kopiert. Außerdem wird diese Datei während des Kopiervorganges auf dem Bildschirm angezeigt.

```
A>PIP LST:=B:LONGPAGE.TXT[FP65]
```

Dieses Kommando schreibt die Datei LONGPAGE.TXT vom Laufwerk B zum Druckgerät. Dabei werden alle enthaltenen

Seitenvorschubzeichen gelöscht und aller 65 Zeichen durch neue ersetzt.

```
B>PIP LST:=PROGRAM.A86[NT8U]
```

Dieses Kommando schreibt die Datei PROGRAM.A86 vom Laufwerk B zum Druckgerät. Auf Grund des N-Parameters wird vor jede Zeile eine Zeilennummer gesetzt. Der T8-Parameter setzt alle 8 Spalten einen Tabulator. Der U-Parameter verlangt, daß alle Kleinbuchstaben als Großbuchstaben ausgedruckt werden.

```
A>PIP PORTION.TXT=LETTER.TXT[SDear Sir^Z QSincerely^Z]
```

Dieses Kommando löst einen Teil der Datei LETTER.TXT von A durch Suchen nach der Zeichenfolge Dear Sir heraus, ehe der Kopiervorgang beginnt. Wird die Zeichenfolge gefunden, wird die Kopie bis zum Auffinden der Zeichenfolge Sincerely nach PORTION.TXT auf A übertragen. Beide Zeichenfolgen sind Bestandteil der Kopie.

```
B>PIP B:=A:*.CMD[VMR]
```

Alle Dateien vom Typ CMD werden von A nach B kopiert. Der V-Parameter veranlaßt PIP, die korrekte Übertragung zu überprüfen. Auf Grund des W-Parameters werden RO-Zieldateien überschrieben. Der R-Parameter bewirkt, daß vom Laufwerk A Dateien mit dem SYS-Attribut gelesen werden.

## 4.14. Kommando REN (Umbenennen)

Syntax:

```
REN {d:}newname{.typ}=oldname{.typ}
```

Typ:

Residentes Kommando

Anwendung:

Mit dem REN-Kommando wird der Name einer Datei geändert, der im Verzeichnis einer Platte enthalten ist. Der Dateiname oldname bezeichnet eine auf der Platte existierende Datei.
Der Dateiname newname darf nicht im Verzeichnis dieser Platte enthalten sein. Das REN-Kommando gibt der Datei mit dem Namen oldname den neuen Dateinamen newname.

REN fertigt keine Dateikopie an. REN ändert nur den Dateinamen.

Bei fehlender Laufwerkskennung bezieht sich REN auf das Standardlaufwerk.

Zur nähereren Spezifikation von newname ist die Angabe einer Laufwerkskennung möglich. Wenn beide Dateispezifikationen eine Laufwerkskennung enthalten, muß diese identisch sein.

Wenn die durch oldname spezifizierte Datei nicht existiert, zeigt REN auf dem Bildschirm an:

```
NO FILE
```

Beispiele:

```
A>REN NEWASM.A86=OLDFILE.A86
```

Die Datei OLDFILE.A86 wird auf dem Laufwerk A im Verzeichnis in NEWASM.A86 umbenannt.

```
B>REN A:X.PAS=Y.PLI
```

Die Datei Y.PLI erhält den neuen Namen X.PAS. Er befindet sich im Verzeichnis der Platte im Laufwerk A.

```
A>REN B:NEWLIST=B:OLDLIST
```

Die Datei OLDLIST auf B erhält den neuen Namen NEWLIST. Da die zweite Laufwerkskennung B: bereits beim neuen Namen spezifiziert ist, kann sie in diesem Beispiel entfallen. Die folgende Kommandozeile hat die gleiche Wirkung:

```
A>REN B:NEWLIST=OLDLIST
```

## 4.15. Kommando STAT (Statusausgabe und -änderung)

Syntax:

```
STAT
STAT d:=RO
STAT filespec {RO|RW|SYS|DIR|SIZE}
STAT {d:}DSK:|USR:
STAT VAL:|DEV:
```

Typ:

Transientkommandos

Anwendung:

Die verschiedenen Formen des STAT-Kommandos geben Informationen über Plattenlaufwerke, Dateien und Geräte, die mit dem Rechner verbunden sind. Mit STAT werden die Datei- und Laufwerkattribute geändert. Die Zuweisung von physischen zu logischen Geräten kann ebenfalls mit STAT erfolgen.

Zu beachten ist, daß die wahlweisen Angaben für die Dateispezifikation in eckige Klammern [] eingeschlossen werden können, ihnen ein ¤-Zeichen vorangestellt oder auf Begrenzer verzichtet werden kann.

Mit dem Attribut RW befindet sich das Laufwerk im Read-Write-Status. Von diesem Laufwerk können Daten sowohl gelesen als auch dorthin geschrieben werden.

Mit dem Attribut RO befindet sich das Laufwerk im Read-Only-Status. Von diesem Laufwerk können Daten nur gelesen werden. Ein Schreiben ist nicht möglich.
Laufwerke sind standardmäßig im RW-Status und erhalten den RO-Status, sobald der Statuswechsel auf RO erfolgt bzw. eine Platte gewechselt und die Eingabe eines CTRL/C vergessen wurde.

### 4.15.1. Setzen eines Laufwerkes auf Read-Only-Status

Syntax:

STAT d:=RO

Anwendung:

Das Kommando setzt ein Laufwerk in den RO-Status. CTRL/C setzt das Laufwerk in den RW-Status zurück.

Beispiel:

```
A>STAT B:=RO
```

Das Laufwerk B erhält den RO-Status.

### 4.15.2. Freier Speicherraum auf der Platte

Syntax:

STAT{d:}

Anwendung:

STAT ohne Parameter zeigt die noch freiverfügbare Plattenkapazität aller angeschlossenen Platten an, d.h. jener Platten, die beim letzten SCP-Start eingegliedert wurden.
Ist die freie Kapazität einer bestimmten Platte zu ermitteln, wird im Kommando die Laufwerkskennung angegeben.

Ist ein Laufwerk, dessen Kennung im Kommando angegeben wurde, nicht im Zustand on-line, bringt SCP dieses Laufwerk in den on-line-Status. Diese Form des STAT-Kommandos zeigt die Informationen auf dem Bildschirm wie folgt an:

```
d:RW, Free Space: nK
```

d ist der Laufspezifikator und n die Anzahl des verbleibenden Speicherraumes in K Byte.

Beispiele:

```
A>STAT
```

Angenommen, es sind zwei Laufwerke mit aktiven Platten angeschlossen, wobei A 16K (16384) Bytes und B 32K (32728) Bytes freien Speicherraum besitzen. A ist mit RW und B mit RO gekennzeichnet, dann erscheint aus dem Bildschirm die Ausschrift:

```
A: RW, Free Space: 16K
B: RO, Free Space: 32K
```

```
A>STAT B:
```

Angenommen, B ist auf RO gesetzt und enthält 98K Byte freien Speicherraum für Programme und Daten, dann erscheint auf dem Bildschirm folgende Mitteilung:

B: RO, Free Space: 98K

Bemerkung:

Die freie Plattenkapazität wird nur dann korrekt angezeigt, wenn die betreffende Platte ordnungsgemäß eingegliedert wurde! (CTRL/C nach Diskettenwechsel)

### 4.15.3. Dateibelegung und Zugriffsmodus

Syntax:

STAT filespec{SIZE}

Anwendung:

Diese Form des STAT-Kommandos zeigt die Größe der spezifizierten Datei in K Byte an. Außerdem wird der Zugriffsmodus dieser Datei ausgegeben. STAT akzeptiert Dateigruppensymbole im Dateinamen und Dateityp. Wird ein solches Sonderzeichen in der Dateispezifikation angegeben, zeigt STAT eine alphabetisch geordnete Liste von Dateien an, die mit dieser Spezifikation identifiziert werden.

Der auf die Dateispezifikation folgende S-Parameter kann in eckige Klammern [] gesetzt werden oder nach einem ¤-Zeichen stehen. Ein Begrenzer kann aber auch entfallen. SCP unterstützt vier Dateizugriffsmodi:

RO Die Datei besitzt ein Read-Only-Attribut. Von ihr können Daten nur gelesen werden. Das Ändern der Datei ist nicht möglich.

RW Die Datei besitzt ein Read-Write-Attribut. Ein Datentransport von und zur Datei ist erlaubt.

SYS Die Datei besitzt ein SYS(System)-Attribut. Systemdateien werden mit dem DIR-Kommando nicht angezeigt. Dazu ist DIRS zu verwenden. Mit dem STAT-Kommando werden alle Dateien angezeigt, einschließlich der mit SYS-Attribut. Das STAT-Kommando zeigt SYS-Dateien in Klammern an.

DIR Die Datei besitzt das DIR(Verzeichnis)-Attribut. Sie wird vom DIR-Kommando angezeigt.

Eine Datei besitzt entweder das RO- oder das RW-Attribut sowie entweder das SYS- oder das DIR-Attribut. Standardmäßig, falls nicht mit dem STAT-Kommando verändert, hat eine Datei die Attribute RW und DIR.

Das STAT-Kommando gibt eine Liste von Dateien aus, deren Kopfzeile folgende Bedeutung hat:

- Die erste Spalte zeigt die Satzanzahl an, die die Datei belegt. Jeder Satz ist 128 Byte lang. Die Satzanzahl wird unter der Bezeichnung Recs. aufgelistet.
- In der zweiten Spalte steht die Dateigröße in K Byte. Ein K Byte enthält 1024 Byte. Diese Werte werden unter Bytes aufgelistet.
- Die dritte Spalte enthält die Zahl der Verzeichniseintragungen, die die Datei benutzt. Diese Werte stehen in der Spalte FCBs. FCB (File Control Block/ Dateisteuerblock) ist ein anderer Name für Verzeichniseintragung.

- Die Zugriffsmodi werden unter Attribute angezeigt.
- Die Dateispezifikation, die die Laufwerkskennung, den Dateinamen und Dateityp enthält, wird unter Name aufgelistet.

Ist eine Laufwerkspezifikation vorhanden und das entsprechende Laufwerk nicht aktiv, wird es von SCP in den aktiven Status gesetzt.
Mit der Angabe SIZE wird durch STAT die virtuelle Dateigröße aller Dateien angezeigt. Die virtuelle und die reale Dateigröße sind für sequentielle Dateien identisch, können aber für Dateien im Direktmodus unterschiedlich sein. Wird der Parameter SIZE angegeben, erscheint auf dem Bildschirm eine zusätzliche Spalte. Die Werte dieser Spalte entsprechen der Summe belegter und leerer Sätze der Datei.

Als Folge des Kommandos STAT *.* führt STAT eine Überprüfung des Verzeichnisses aus, um zu verhindern, daß zwei Dateien den gleichen Plattenraum belegen, d.h. eine Datei mit einer anderen Datei gemeinsame Teile auf der Platte belegt. Erkennt STAT diesen Zustand, wird folgende Mitteilung auf dem Bildschirm ausgegeben:

```
Bad Directory on d:
Space Allocation Conflict:
User nn d:filename.typ
```

STAT gibt die Nutzernummer und die Dateispezifikation der Datei aus, die diese Doppelbelegung besitzt. Es können mehrere Dateien betroffen sein. Die angezeigten Dateien sind zu löschen, und anschließend ist CTRL/C einzugeben.

STAT führt immer dann eine Überprüfung des Verzeichnisses aus, wenn im Kommando ein Sonderzeichen angegeben wird.

Beispiele:

```
A>STAT MY*.*
```

Mit diesem Kommando wird verlangt, die Eigenschaften aller Dateien anzuzeigen, die mit MY beginnen, wobei der Dateityp keine Rolle spielt. Angenommen, die folgenden drei Dateien erfüllen die Spezifikation, dann erscheint auf dem Bildschirm:

```
Drive A:                                  User 0
Recs    Bytes    FCBs    Attributes       Name
  16      2K       1       Dir RW       A:MYPROG  .A86
   8      1K       1       Dir RO       A:MYTEST  .DAT
  32     18K       2       Sys RO       A:MYTRAN  .CMD
Total:   21K       4
A: RW, Free Space: 90K
```

```
A>STAT MY*.* SIZE
```

Dieses Kommando bewirkt die gleiche Reaktion wie das vorherige Kommando, zusätzlich wird die Spalte Size angezeigt. Falls die Datei MYTEST.DAT im Direktmodus von Satznummer 8 bis 15 geschrieben wurde und die ersten 8 Sätze leer sind, so beträgt die virtuelle Größe 16 Sätze, obwohl von der Datei nur 8 Sätze verbraucht werden. Auf dem Bildschirm ist dann folgende Ausschrift zu sehen:

```
Drive A:                                              User 0
Size    Recs     Bytes      FCBs    Attributes        Name
  16      16        2K         1      Dir RW        A:MYPROG  .A86
  16       8        1K         1      Dir RO        A:MYTEST  .DAT
  32      32       18K         2      Sys RO        A:MYTRAN  .CMD
Total:  21K      4
A: RW, Free Space: 90K
```

### 4.15.4. Setzen der Dateizugriffsmodi (Attribute)

Syntax:

```
STAT filespec RO|RW|SYS|DIR
```

Anwendung:

Mit diesem Kommando wird der Dateizugriffsmodus für eine oder mehrere Dateien gesetzt. Der auf die Dateispezifikation folgende Parameter kann in eckige Klammern gesetzt oder ihm kann ein =-Zeichen vorangestellt werden. Ein Begrenzer kann aber auch entfallen.

Die vier Zugriffsmodi , die bereits beschrieben wurden, sind:

```
RO
RW
SYS
DIR
```

Ist das angegebene Laufwerk inaktiv, wird es von SCP zuerst in den on-line-Status gesetzt.

Eine Datei kann entweder das RO- oder das RW-Attribut, aber nicht beide besitzen. Ebenso kann eine Datei entweder das SYS- oder das DIR-Attribut, aber nicht beide besitzen.

Beispiele:

```
A>STAT LETTER.TXT RO
```

Der Zugriffmodus für die Datei LETTER.TXT auf der Platte im Laufwerk A wird auf RO gesetzt. Wenn die Datei vorhanden ist, erscheint auf dem Bildschirm:

```
LETTER.TXT set to RO
```

```
B>STAT A:*.CMD SYS
```

Das Kommando setzt den Zugriffsmodus aller Dateien vom Typ CMD auf der Platte in A auf SYS. Angenommen, es befinden sich die drei Kommandodateien PIP, ED und ASM auf A, dann erscheint folgende Mitteilung auf dem Bildschirm:

```
PIP.CMD set to SYS
ED.CMD set to SYS
ASM.CMD set to SYS
```

### 4.15.5. Anzeige des Plattenstatus

Syntax:

STAT{d:}DSK:

Anwendung:

Dieses Kommando zeigt interne Informationen aller angeschlossenen Platten des Systems an. Wenn ein Laufwerk spezifiziert wird, dann wird es in den on-line-Status gesetzt.

Die mit diesem Kommando gezeigten Informationen sind für fortgeschrittene Programmierer sinnvoll. Im täglichen Gebrauch von SCP werden sie nicht benötigt.

Beispiele:

```
A>STAT DSK:
```

Über das Laufwerk A werden folgende Informationen angezeigt (die Werte für n sind Ziffern):

```
   A: Drive Characteristics
nnnn: 128 Bytes Record Capacity
nnnn: Kilobyte Drive Capacity
nnnn: 32 Bytes Directory Entries
nnnn: Checked Directory Entries
nnnn: 128 Bytes Records/Directory Entry
nnnn: 128 Bytes Records/Block
nnnn: 128 Bytes Records/Track
nnnn: Reserved Tracks
```

```
A>STAT B:DSK:
```

Dieses Kommando gibt die im vorhergehenden Beispiel gezeigten Informationen für das Laufwerk B aus.

### 4.15.6. Anzeige der Nutzernummer der Bereiche mit aktiven Dateien

Syntax:

STAT{d:}USR:

Anwendung:

Mit diesem Kommando können die Nutzernummern bestimmt werden, die die Dateien auf der Platte im spezifizierten Laufwerk haben. Nutzernummern werden Dateien zugewiesen, die unter SCP erstellt wurden.

Beispiel:

```
A>STAT USR:
```

Dieses Kommando zeigt die Nutzernummer der aktiven Dateien auf der Platte in A an.

### 4.15.7. Anzeige der STAT-Kommandos und Gerätenamen

Syntax:

STAT VAL:

Anwendung:

STAT VAL: ist die allgemeine Form der STAT-Kommandos. Es gibt an, welche physische Gerätenamen jedem der vier SCP-logischen Geräte zugewiesen werden können.

Beispiel:

```
A>STAT VAL:

STAT 1.0
Read Only Disk: d:=RO
Set Attribute: d:filename.typ[ro] [rw] [sys] [dir]
Disk Status: DSK: d:DSK:
User Status: USR: d:USR:
Iobyte Assign:
CON: = TTY: CRT: BAT: UC1:
AXI: = TTY: PRT: UR1: UR2:
AXO: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
A>
```

### 4.15.8. Anzeige und Setzen physischer und logischer Gerätezuordnungen

Syntax:

STAT DEV:
STAT logical device: = physical device:

Anwendung:

STAT DEV: zeigt die aktuellen Zuweisungen für die vier SCP-logischen Gerätenamen CON:, AXI:, AXO: und LST: an. Sind diese Zuweisungen zu ändern, wird die zweite Kommandoform genutzt. Das Kommando STAT VAL: zeigt die möglichen physischen Gerätenamen an, die jedem logischen Gerätenamen zugewiesen werden können (siehe Beispiel zu STAT VAL: unter dem Titel Iobyte Assign).

Wird einem physichen Gerät ein logisches Gerät zugewiesen, trägt STAT Werte von 0 bis 3 für das logische Gerät in das sogenannte Iobyte ein.

Jedes der ausgelisteten physischen Geräte kann dem passenden logischen Gerätenamen zugewiesen werden. Die Zuweisung bleibt wirkungslos, wenn nicht die richtige E/A-Steuerung des Rechners mit den zugehörigen Kabeln angeschlossen und nicht der richtige Driver für die E/A für das betreffende physische Gerät Systembestandteil ist.

Die physischen Gerätedriver sind im BIOS zu implementieren. Das Iobyte wird gelesen, ausgewertet und der der logischen Ausgaberoutine entsprechende Driver angesprungen. Weitere Informationen über die Handhabung externer physicher Geräte sind der Anleitung

für den Systemprogrammierer zu entnehmen.

Beispiele:

A>STAT CON: = CRT:

Das physische Gerät mit dem Namen CRT: wird dem logischen Eingabegerät mit dem Namen CON: zugewiesen, das sich im allgemeinen auf das Terminal (Tastatur) bezieht.

A>STAT LST: = LPT:

Das physische Gerät mit dem Namen LPT: wird dem logischen Gerät mit dem Namen LST: zugewiesen, das sich im allgemeinen auf den Drucker bezieht.

## 4.16. Kommando SUBMIT (Stapelverarbeitung)

Syntax:

SUBMIT filespec{parameters...}

Typ:

Transientprogramm

Anwendung:

Das SUBMIT-Dienstprogramm ermöglicht es, einen Satz von Kommandos zusammenzustellen, der durch SCP automatisch abgearbeitet wird. Normalerweise werden Kommandos Zeile für Zeile eingegeben. Ist die gleiche Kommandofolge mehrere Male abzuarbeiten, ist es leichter, diese Kommandos unter Ausnutzung von SUBMIT in Stapelverarbeitung ablaufen zu lassen. Dazu muß eine Datei erstellt werden, die diese Kommandofolge enthält. Diese Datei wird durch ihren Namen identifiziert und muß den Typ SUB haben. Wird das SUBMIT-Kommando gegeben, liest dieses Dienstprogramm die Datei entsprechend der Dateispezifikation und sorgt für ihre Interpretation durch SCP.

Die Datei vom Typ SUB kann alle zulässigen SCP-Kommandos enthalten. SUBMIT-Parameter können in die SUB-Datei einbezogen werden. Diese werden durch Werte ersetzt, die als Parameter des Kommandos eingegeben wurden.

SUBMIT-Parameter haben die allgemeine Form ¤n (n = 0 bis 9):

¤0
¤1
¤2
¤3
¤4
¤5
¤6
¤7
¤8
¤9

Die Parameter können überall in den Kommandozeilen in der SUB-Datei angegeben werden.

Das SUBMIT-Dienstprogramm liest die dem Kommandoschlüsselwort folgende Datei gemäß filespec und benutzt die aktuellen Parameter in der Kommandozeile, um die allgemeinen Parameter in der SUB-Datei zu ersetzen. Ist diese Substitution erfolgt, sendet SUBMIT die Datei Zeile für Zeile an das SCP, als würden die Zeilen einzeln eingegeben.

Jeder Parameter im Kommando ist eine Folge von alphanumerischen und/oder speziellen Zeichen. Die Parameter werden durch ein oder mehrere Leerzeichen getrennt.

Der Parameter ¤0 wird durch den in filespec angegebenen Dateinamen ersetzt. Der erste Parameter ersetzt ¤1, der zweite ¤2 usw. Werden weniger Parameter als in der SUB-Datei erwartet angegeben, so werden die restlichen in der SUB-Datei als gelöscht betrachtet. Werden mehr als notwendig angegeben, so werden die überzähligen Parameter ignoriert.

SUBMIT erstellt eine temporäre Datei ¤¤¤.SUB, die die substituierten Kommandos enthält. Die Stapelverarbeitung endet nach dem Lesen der letzten Zeile der SUB-Datei.

CTRL-Break stoppt den SUBMIT-Prozeß. Die Stapelverarbeitung kann auch durch Eingabe eines beliebigen Zeichens abgebrochen werden, wenn die SCP-Anforderung (A>) auf dem Bildschirm sichtbar ist. Die Datei ¤¤¤.SUB wird automatisch gelöscht, wenn SCP alle Kommandozeilen abgearbeitet hat. SUBMIT-Dateien dürfen keine geschachtelten Kommandos enthalten, aber das letzte Kommando in einer SUB-Datei kann ein SUBMIT-Kommando sein, das eine andere SUB-Datei aktiviert. Um ein ¤-Zeichen in die SUB-Datei einfügen zu können, sind zwei (¤¤) einzugeben. Zu beachten ist, daß die SUB-Datei keine Kommandozeile enthält, die null Zeichen lang ist, da in diesem Falle alle Kommandozeilen, die vor dieser Zeile stehen nicht abgearbeitet werden.

Beispiele:

```
A>SUBMIT SUBFILE
```

Die Datei SUBFILE.SUB befindet sich auf der Platte in A und enthält folgende Kommandos:

```
DIR*.CDM
ASM86 X.¤¤¤SB
PIP LST:=X.LST[T8D80]
```

SUBMIT sendet die Kommandofolge, die die Datei SUBFILE.SUB enthält, an das SCP zur Bearbeitung. SCP führt zuerst das DIR-Kommando aus, dann wird die Datei X.A86 assembliert und schließlich das PIP-Kommando ausgeführt.

```
A>SUBMIT B:ASMCOM X 8 D80 SZ
```

```
Zuweisung:    X  -->  ¤1
              8  -->  ¤2
            D80  -->  ¤3
             SZ  -->  ¤4
```

Die Datei ASMCOM.SUB befindet sich auf B: und enthält folgende Kommandos:

```
ERA ¤1.BAK
ASM86 ¤1 ¤¤¤4
PIP LST:=¤1.LST[T¤2 ¤3 ¤5]
```

SUBMIT liest diese Datei und substituiert wie folgt:

```
ERA X.BAK
ASM86 X ¤SZ
PIP LST:=X.LST[T8 D80]
```

Diese Kommandos werden von oben nach unten von SCP abgearbeitet.

## 4.17. Kommando TYPE (Anzeige von Textdateien)

Syntax:

TYPE {d:}filename{.typ}

Typ:

Residentes Kommando

Anwendung:

Mit dem TYPE-Kommando werden Inhalte von Textdateien auf dem Bildschirm sichtbar gemacht.

Tabulatoren, die die auszugebende Datei enthält, werden auf dem Bildschirm aller 8 Spalten gesetzt.

Mit jeder beliebigen Taste der Tastatur kann die TYPE-Anzeige unterbrochen werden.

Es ist abzusichern, daß dem TYPE-Kommando nur Textdateien angeboten werden.

Ist die spezifizierte Datei nicht auf der on-line-Platte vorhanden, gibt TYPE folgende Mitteilung über den Bildschirm aus:

```
NO FILE
```

Soll die Datei gleichzeitig auf dem Drucker und dem Bildschirm ausgegeben werden, ist CTRL/P vor dem RETURN des TYPE-Kommandos einzugeben. Ein erneutes CTRL/P schaltet das Echo auf dem Drukker wieder aus.

Beispiele:

```
A>TYPE MYPROG.A86
```

Der Inhalt der Datei MYPROG.A86 wird auf dem Bildschirm angezeigt.

```
A>TYPE B:THISFILE
```

Der Inhalt der Datei THISFILE auf der Platte in B wird zum Bildschirm übertragen.

## 4.18. Kommando USER (Nutzernummer)

Syntax:

USER {number}

Typ:

Residentes Kommando

Anwendung:

Mit dem Kommando USER wird die laufende Nutzernummer angezeigt und verändert. Das Plattenverzeichnis kann in unterschiedliche Gruppen eingeteilt werden, die jeweils zu einer Nutzernummer gehören.

Beim Start des SCP ist die aktuelle Nutzernummer 0. Auf die mit einer bestimmten Nutzernummer generierten Dateien kann von anderen Nutzernummern nicht zugegriffen werden. Ausnahmen bilden das PIP-Kommando oder Dateien, denen mit dem STAT-Kommando das Systemattribut (SYS) zugewiesen wurde (siehe Parameter in Abschnitt 4.13.6.).
Mit dem Kommando

```
USER
```

wird die aktuelle Nutzernummer angezeigt.
Mit dem Kommando

```
USER number
```

wird die aktuelle Nutzernummer auf number (0...15) gesetzt.
Das Kommando

```
STAT USR:
```

erzeugt die Liste der Nutzernummer und der ihnen zugeordneten Dateien.

Beispiele:

```
A>USER
```

Dieses Kommando zeigt die aktuelle Nutzernummer.

```
A>USER 3
```

Die aktuelle Nutzernummer wird auf 3 gesetzt.

## 4.19. LDCOPY-Kommando (SCP-1700-Systemspuren kopieren)

Syntax:

LDCOPY oder LDCOPY d:filespec

Typ:

Transientes Kommando

Anwendung:

LDCOPY kopiert den SCP-1700-Lader (LDSCP.CMD) entweder von den Systemspuren einer existierenden SCP-1700-Systemdiskette oder von einer Datei auf eine neue Diskette.
Nach dem Aufruf

```
LDCOPY
```

meldet sich das Programm mit

```
LDCOPY   VER x.y
source drive name
```

und erwartet die Eingabe der Laufwerksbezeichnung (A-P) für die Quellplatte. Alle Eingaben sind mit (CR) abzuschließen. Nach Beendigung des Einlesens der Systemspuren erscheint die Mitteilung

```
function complete
destination drive name (or return to reboot)
```

Es wird die Laufwerksbezeichnung (A-P) der Zielplatte erwartet. Die Beendigung des Schreibvorganges wird wieder angezeigt durch die Ausschrift

```
function complete
```

Nun können weitere Zielplatten angegeben werden oder LDCOPY wird durch die Eingabe von (CR) beendet.
Soll der Lader nicht von einer vorhandenen Systemplatte überspielt, sondern von einer Datei auf eine neue Diskette gebracht werden, ist die Dateispezifikation nach dem Namen anzugeben. Vom Programm aus wird dann nur die Laufwerkskennung (A-P) der Zielplatte angefordert.

Fehler:

Falsche Laufwerksangaben für Ziel- und Quellplatte führen zur Fehlermeldung:

```
invalid drive name
```

Es wird mit einer erneuten Eingabeanforderung fortgesetzt.

Fehler beim Lesen bzw. Schreiben der Systemspuren führen zu den Fehlerausschriften:

```
read error  oder  write error
```

Wird nach dem Programmnamen eine nicht vorhandene Dateispezifikation angegeben, kommt vom Programm die Mitteilung:

```
no source file on disk
```

# 5. SCP-Editor ED

## 5.1. Vorbemerkungen zum ED

Bei fast allen Arbeiten mit dem Rechner ist es notwendig, Daten einzugeben. Die für diese Aufgabe am meisten genutzten Programme sind die Editoren. Sie übertragen Tastatureingaben zu einer Plattendatei. Der SCP-Editor heißt ED. Mit Hilfe von ED können Textdateien erstellt und geändert werden.
Die korrekte Kommandosyntax für das Aufrufen des SCP-Editors wird im Abschnitt 5.2. beschrieben. Nach dem Start des ED werden Kommandos eingegeben, die den Text von der Plattendatei zum Editieren in den Speicher übertragen. Im Abschnitt 5.3. werden diese detailliert beschrieben und grundlegende Kommandos zur Textübertragung vorgestellt. Abschnitt 5.4. behandelt ausführlich die Kommandos zum Editieren einer Datei. Abschnitt 5.5. beschreibt, wie durch Kombination von Basiskommandos effektiver editiert werden kann. Obwohl jede Datei mit den Basiskommandos des ED aufbereitet werden kann, stellt ED weitere Kommandos bereit, die kompliziertere Editierfunktionen darstellen. Diese werden im Abschnitt 5.6. beschrieben. Während eines Editiervorganges kann ED mit zwei verschiedenen Typen von Fehlermeldungen reagieren. Im Abschnitt 5.7. sind diese Fehlerausschriften erläutert, verbunden mit Beispielen für die Fehlerursachen und deren Beseitigung.

## 5.2. Start des ED

Syntax:

```
ED filespec {d:|filespec}
```

Um ED zu starten, muß nach der Systemmeldung des SCP das Kommandoschlüsselwort ED eingegeben werden. Diesem folgt eine Dateispezifikation ohne Sonderzeichen, wie z. B.:

```
A>ED MYFILE.TEX
```

Die Dateispezifikation, im obigen Beispiel MYFILE.TEX, bezeichnet eine Datei, die editiert oder erstellt werden soll. Eine Laufwerkskennung kann angegeben werden, sie ist aber nicht notwendig, wenn sich die zu editierende Datei auf einer Platte im Standardlaufwerk befindet. Wahlweise kann der Dateispezifikation eine Laufwerkskennung folgen, wie es im nächsten Beispiel gezeigt wird.

```
A>ED MYFILE.TEX B:
```

Mit diesem Kommando eröffnet ED die zu editierende Datei MYFILE.TEX auf der Platte im Laufwerk A, sendet aber alle Daten zur editierten Datei auf B.

Wahlweise kann die editierte Datei mit einem anderen Namen versehen werden, wie im folgenden Beispiel:

```
A>ED MYFILE.TEX YOURFILE.TEX
```

Die zweite Datei darf noch nicht existieren, sonst gibt ED die Mitteilung aus:

```
Output File Exists, Erase It
```

und wird abgebrochen.

Die ED-Meldung * erscheint auf dem Bildschirm, wenn ED eine Kommandoeingabe erwartet:

```
A>ED MYFILE.TEX
:*
```

Falls die angegebene Datei auf der aktuellen Platte nicht existiert, erstellt ED eine neue Datei und gibt folgende Mitteilung aus:

```
NEW FILE
:*
```

Vor dem Editieren sollte mit dem STAT-Kommando geprüft werden, ob auf der Platte noch genügend freier Speicherraum vorhanden ist. Der unbenutzte Plattenraum muß so groß wie die zu editierende Datei sein oder größer, falls zusätzliche Zeichen in die Datei eingefügt werden sollen. Wenn ED erkennt, daß eine Platte oder ein Verzeichnis voll ist, bricht es den Editiervorgang ab. Durch Löschen nichtbenötigter Dateien muß Platz geschaffen oder die editierte Datei auf eine andere Platte übertragen (siehe Abschnitt 5.7.) werden.

## 5.3. ED-Operationen

Mit ED können Teile von Dateien verändert werden, wenn sich diese Teile in einem Speicherpuffer befinden. Beim Start von ED ist der Speicherpuffer leer. Infolge eines Kommandos liest ED Teile der Quelldatei, z.B. MYFILE.TEX, in den Speicherpuffer zum Editieren ein. Während des Editierens schreibt ED den editierten Text auf eine temporäre Arbeitsdatei MYFILE.¤¤¤.

Nach Beendigung des Editiervorgangs schreibt ED den Inhalt des Speicherpuffers und anschließend den restlichen Text aus der Quelldatei in die temporäre Datei. Danach ändert ED den Namen der Datei MYFILE.TEX in MYFILE.BAK, damit kann auch weiterhin zum Originaltext zugegriffen werden. ED benennt dann die neue Datei MYFILE.¤¤¤ in MYFILE.TEX um.

Wenn ED mit zwei Dateispezifikationen aufgerufen wird, einer Eingabe- und einer Ausgabedatei, nennt ED die Eingabedatei nicht in den Typ .BAK um. Deshalb kann sie eine Read-Only-Datei sein oder sich auf einer schreibgeschützten Platte befinden, wenn die Ausgabedatei zu einer anderen Platte geschrieben wird.

Die folgende Tabelle zeigt eine Übersicht über die drei Basiskommandos für die Textübertragung, mit denen der Editiervorgang begonnen, Text in eine temporäre Datei geschrieben und die Arbeit des ED beendet werden können.

Tabelle 10: Textübertragungskommandos

| Kommando | Ergebnis |
|---|---|
| nA | Anfügen von n unbearbeiteten Quellzeilen aus der Quelldatei an den gefüllten Teil des Speicherpuffers. |
| nW | Schreiben der ersten n Zeilen des Speicherpuffers zur temporären Datei. |
| E | Beenden von ED. Kopieren des gesamten gepufferten Textes zur temporären Datei und Kopieren der unbearbeiteten (restlichen) Quellzeilen in die temporäre Datei. Umbenennen der Dateien. |

### 5.3.1. Einlesen von Text in den Puffer

Wenn ED gestartet wird und der Speicherpuffer leer ist, kann mit dem Kommando A (Append) Text in den Speicherpuffer übertragen bzw. zum vorhandenen Text hinzugefügt werden.

ED numeriert die Zeilen des Textes im Speicherpuffer. Der Doppelpunkt, der nach dem Start des ED erscheint, zeigt an, daß diese Zeilennumerierung aktiviert ist. Mit dem Kommando -V nach der ED-Meldung kann diese Numerierung ausgeschaltet werden. Zeilennummern erscheinen nur auf dem Bildschirm, sie sind nicht Bestandteil der editierten Ausgabedatei.

#### Kommando A (Append)

Das A-Kommando kopiert Zeilen aus einer vorhandenen Quelldatei in den Speicherpuffer. Es hat die Form:

nA

wobei n die Anzahl der unbearbeiteten Quellzeilen ist, die in den Puffer zu übertragen ist. Wenn für n das Zeichen # angegeben wird, werden 48253 Bytes als Speicherpuffer reserviert. Wenn die Quelldatei nicht größer ist, kann das Kommando #A zu Beginn des Editierens für das Einlesen der gesamten Quelldatei benutzt werden.
Wird für n eine 0 angegeben, dann werden so viele unbearbeitete Quellzeilen in den Puffer gelesen, bis dieser etwa zur Hälfte gefüllt ist. Wird kein n angegeben, dann wird nur eine Quellzeile eingelesen.

### 5.3.2. Abschließen des Editiervorganges

Zum Sichern des editierten Textes können das Kommando W (Write) und das Kommando E (Exit) benutzt werden. Mit dem W-Kommando werden Textzeilen aus dem Speicherpuffer zur neuen Datei geschrieben, ohne die Arbeit mit ED abzubrechen. Ein E-Kommando rettet den Pufferinhalt und alle unbearbeiteten Zeilen der Quelldatei in die neue Datei, benennt die Dateien um und bricht den ED ab.

#### Kommando W (Write)

Das W-Kommando schreibt Zeilen aus dem Puffer zur neuen Datei. Die Form des W-Kommandos ist:

```
nW
```

n ist die Zahl der Zeilen, die aus dem Puffer in die neue Datei geschrieben werden. Diese Zeilen werden in der neuen Datei an das Ende des schon eingetragenen Textes angefügt. Wenn n größer als 0 ist, schreibt ED n Zeilen ab Pufferanfang an das Ende des schon vorhandenen Textes in die neue Datei. Wird für n eine 0 angegeben, schreibt ED solange Zeilen zur neuen Datei, bis der Puffer zur Hälfte leer ist. Mit dem 0W-Kommando kann im Puffer Platz für weitere Quellzeilen geschaffen werden. Die Anzahl von Zeilen, die zu schreiben sind, kann mit dem Kommando 0V bestimmt werden, das den freien Platz im Puffer ermittelt:

```
1:*0V

25000/48253
1:*
```

Das Beispiel zeigt, daß die Gesamtgröße des Puffers 48253 Bytes beträgt und noch 25000 Bytes im Puffer frei verfügbar sind.

Nach der Ausführung eines W-Kommmandos muß das H-Kommando eingegeben werden, wenn mit ED erneut zu den geretteten Zeilen zugegriffen werden soll.

#### Kommando E (Exit)

Ein E-Kommando bewirkt den normalen Abbruch der Arbeit mit ED. Die Form des E-Kommandos ist:

```
E
```

gefolgt von RETURN.

Bei Eingabe eines E-Kommandos werden zuerst alle Datenzeilen aus dem Puffer und der Rest der Quelldatei zur neuen Datei geschrieben. Existiert eine Datei vom Typ BAK, so löscht ED diese und benennt die Originaldatei in den Typ BAK um. Schließlich erhält die neue Datei mit dem Typ ¤¤¤ den ursprünglichen Typ der Originaldatei und die Steuerung geht an das SCP. Die Wirkungsweise des E-Kommandos zeigt, daß es ungünstig ist, eine BAK-Datei zu editieren, da ED mit dem E-Kommando die Originaldatei löschen würde. Um das zu verhindern, muß der BAK-Datei ein anderer Typ zugewiesen werden, bevor das Editieren beginnt.

Jedes Kommando, das die Arbeit eines Editiervorganges beendet, muß als einziges Kommando auf einer Zeile stehen.

## 5.4. Basiskommandos zum Editieren

Dieser Abschnitt beschreibt die Basiskommandos zum Editieren einer Datei. ED behandelt eine Datei wie eine Kette von Zeichen, die in Zeilen zusammengefaßt sind. ED zeigt und editiert Zeichen und Zeilen in Abhängigkeit von einem imaginären Zeichenzeiger CP (Character Pointer). Während des Editiervorganges muß die Position des CP im Speicherpuffer bekannt sein, um den CP durch Kommandos richtig bewegen und eine Datei editieren zu können.

Die folgenden Kommandos bewegen den CP oder zeigen Text in der Umgebung des CP an. Diese ED-Kommandos bestehen aus einem numerischen Argument und einem einzelnen Buchstaben, dem ein RETURN folgen muß. Das numerische Argument n bestimmt die Anzahl, wie oft das Kommando auszuführen ist. Es gibt jedoch vier Spezialfälle, die bei der Betrachtung des numerischen Argumentes zu berücksichtigen sind:

- Wird das Argument nicht angegeben, setzt ED das Argument auf 1.
- Wird ein negatives Argument angegeben, wird das Kommando bezogen auf den CP im Puffer rückwärts ausgeführt (das B-Kommando ist eine Ausnahme).
- Wird als numerisches Argument ein # angegeben, führt ED dieses Kommando mit dem auf den CP folgenden Inhalt des Speicherpuffer (48253 Bytes) aus. Dem # kann ein Minuszeichen (-#) vorgesetzt werden, wenn das Kommando rückwärts ausgeführt werden soll.
- ED akzeptiert eine 0 als numerisches Argument nur in bestimmten Kommandos. In einigen Fällen bewirkt das Argument 0, daß das Kommando etwa halb so oft ausgeführt wird, wie es möglich wäre, während es in anderen Fällen die Bewegung des CP verhindert. Die folgende Tabelle faßt die Basiskommandos zum Editieren in alphabetischer Reihenfolge zusammen und enthält die zulässigen Argumente.

Tabelle 11: Basiskommandos zum Editieren

| Kommando | Wirkung |
|---|---|
| B, -B | Bewegen des CP an den Beginn (B) oder das Ende (-B) des Speicherpuffers. |
| nC, -nC | Bewegen des CP im Speicherpuffer um n Zeichen vorwärts (nC) oder rückwärts (-nC). |
| nD, -nD | Löschen von n Zeichen vor (-nD) oder hinter (nD) dem CP. |
| I | Setzen des Eingabemodus. |
| Istring^Z | Eingabe einer Zeichenkette (string). |
| nK, -nK | Löschen von n Zeilen vor dem CP (-nK) oder hinter dem CP (nK). |
| nL, -nL | Bewegen des CP im Speicherpuffer um n Zeilen vorwärts (nL) oder rückwärts (-nL). |
| nT, -nT | Anzeige von n Zeilen vor dem CP (-nT) oder hinter dem CP (nT). |
| n, -n | Bewegen des CP um n Zeilen vor dem CP (-n) oder nach dem CP (n) und Anzeigen der Zielzeile. |

Die folgenden Abschnitte behandeln die Basiskommandos zum Editieren in detaillierter Form. Die Beispiele in diesen Abschnitten illustrieren, wie mit Kommandos die Position des Zeichenzeigers im Speicherpuffer verändert werden kann. Mit den Beispielen im Abschnitt 5.5. wird gezeigt, wie die Kommandos auf dem Bildschirm erscheinen. In den Beispielen in diesen Abschnitten wird das Symbol ^ benutzt, um die Position des CP im Speicherpuffer darzustellen.

### 5.4.1. Positionieren des Zeichenzeigers CP

Dieser Abschnitt beschreibt Kommandos, die den CP bewegen, aber die Zielzeile nicht anzeigen. Obwohl ED in erster Linie zum Erstellen und Editieren von Programmquelldateien benutzt wird, erfolgen die Erläuterungen in den nächsten Abschnitten an einem Textbeispiel, das einfach zu verstehen ist.

#### Kommando B (Beginning/Bottom)

Das B-Kommando bewegt den CP an den Anfang oder das Ende des Speicherpuffers. Die Formen des B-Kommandos sind:

B, -B

-B bewegt den CP an das Ende des Puffers;
B bewegt den CP an den Anfang des Puffers.

#### Kommando C (Character)

Das C-Kommando bewegt den CP um die spezifizierte Anzahl von Zeichen vorwärts oder rückwärts. Die Formen des C-Kommandos sind:

nC, -nC

wobei n die Zahl der Zeichen darstellt, um die der CP zu bewegen ist. Eine positive Zahl bewegt den CP vorwärts in Richtung

Zeilen- und Pufferende. Eine negative Zahl bewegt den CP rückwärts in Richtung Zeilen- und Pufferanfang. n kann auch so groß angegeben werden, daß es den CP auf eine andere Zeile positioniert. Jede Zeile ist von der anderen durch die nicht sichtbaren Zeichen (CR)(LF) getrennt. Diese müssen im Argument des Kommandos berücksichtigt werden. Das Kommando 30C bewegt den CP im folgenden Beispiel auf die nächste Zeile:

```
Emily Dickinson said,(CR)(LF)
"I fin^d ecstasy in living -(CR)(LF)
```

Kommando L (Line)

Das L-Kommando bewegt den CP um die spezifizierte Anzahl von Zeilen. Nach einem L-Kommando zeigt der CP immer auf den Beginn einer Zeile. Die Formen des L-Kommandos sind:

```
nL, -nL
```

wobei n die Zeilenanzahl für die CP-Bewegung ist.

Eine positive Zahl bewegt den CP vorwärts in Richtung Pufferende. Eine negative Zahl bewegt den CP rückwärts in Richtung Pufferanfang. Das Kommando 2L bewegt den CP zwei Zeilen vorwärts und auf den Zeilenanfang:

```
Emily Dickinson said,(CR)(LF)
"I find ecstasy in living -(CR)(LF)
^the mere sense of living(CR)(LF)
```

Das Kommando -L bewegt den CP an den Beginn der vorhergehenden Zeile, wenn der CP ursprünglich in der Mitte einer Zeile stand. Mit dem Kommando 0L wird der CP an den Anfang der aktuellen Zeile positioniert.

Kommando n (Number)

Das n-Kommando bewegt den CP und zeigt die Zielzeile an. Die Formen des n-Kommandos sind:

```
n, -n
```

wobei n die Zeilenzahl für die CP-Bewegung ist. ED bewegt den CP vorwärts oder rückwärts entsprechend der Zeilenanzahl und zeigt nur die Zielzeile an:

```
 Emily Dickinson  said,(CR)(LF)
^"I find ecstasy in living -(CR)(LF)
```

Eine weitere Möglichkeit besteht darin, überhaupt keine Zahl einzugeben. ED positioniert den CP nach Eingabe eines RETURN ohne eine vorgesetzte Zahl auf die nächste Zeile und zeigt diese an:

```
 Emily Dickinson said,(CR)(LF)
^"I find ecstasy in living -(CR)(LF)
```

Analog positioniert auch ein Minuszeichen vor einem RETURN den CP an den Anfang der vorherigen Zeile.

### 5.4.2. Anzeige der Speicherpufferinhalte

ED zeigt den Inhalt des Speicherpuffers an, wenn angegeben wird, welcher Teil des Textes gewünscht wird. Das T-Kommando zeigt Text an, ohne dabei den CP zu bewegen.

#### Kommando T (Type)

Das T-Kommando überträgt eine spezifizierte Anzahl von Zeilen ab dem CP zum Bildschirm. Die Formen des T-Kommandos sind:

```
nT, -nT
```

wobei n die Anzahl der anzuzeigenden Zeilen ist. Wird eine negative Zahl eingegeben, zeigt ED n Zeilen vor dem CP an. Bei einer positiven Zahl werden n Zeilen hinter dem CP angezeigt. Wird keine Zahl n spezifiziert, erfolgt die Ausgabe der Zeile ab CP bis Zeilenende. Der CP verbleibt auf seiner Position. Dabei spielt es keine Rolle, wieviel Zeilen angezeigt werden. Befindet sich z.B. der CP am Pufferanfang und wird die Ausgabe von vier Zeilen (4T) verlangt, so erscheinen vier Zeilen auf dem Bildschirm, aber der CP steht am Beginn der Zeile 1.

```
^Emily Dickinson said,(CR)(LF)
 "I find ecstasy in living -(CR)(LF)
 the mere sense of living(CR)(LF)
 is joy enough."(CR)(LF)
```

Befindet sich der CP zwischen zwei Zeichen in der Mitte einer Zeile, dann zeigt ein T-Kommando ohne Spezifikation der Zeilenanzahl nur die Zeichen zwischen dem CP und dem Zeilenende an, und der CP verbleibt an seiner Position im Speicherpuffer:

```
"I find ec^stasy in living -(CR)(LF)
```

Während der Textanzeige infolge eines T-Kommandos kann jederzeit durch Eingabe von CTRL/S die Anzeige angehalten und mit CTRL/Q wieder fortgesetzt werden. Sehr lange Textanzeigen können mit CTRL/C abgebrochen werden.

### 5.4.3. Löschen von Zeichen

#### Kommando D (Delete)

Das D-Kommando löscht eine spezifizierte Anzahl von Zeichen und besitzt die Formen:

```
nD, -nD
```

wobei n die Anzahl der zu löschenden Zeichen angibt. Wird kein n spezifiziert, löscht ED das rechts vom CP stehende Zeichen. Ein positives n löscht entsprechend viele Zeichen rechts vom CP in Richtung zum Pufferende. Ein negatives n löscht entsprechend viele Zeichen links vom CP in Richtung Pufferanfang. Wenn der CP im Puffer wie im folgenden Beispiel positioniert ist:

```
Emily Dickinson said,(CR)(LF)
"I find ecstasy in living-(CR)(LF)
the mere sense of living(CR)(LF)
is joy ^enough."(CR)(LF)
```

löscht das Kommando 6D sechs Zeichen hinter dem CP, und im Puffer zeigt sich folgendes Ergebnis:

```
Emily Dickinson said,(CR)(LF)
"I find ecstasy in living -(CR)(LF)
the mere sense of living (CR)(LF)
is joy ^."(CR)(LF)
```

Mit einem D-Kommando können auch die Zeichen (CR)(LF) zwischen zwei Zeilen gelöscht werden, um diese zu einer Zeile zusammenzufügen. Zu beachten ist, daß (CR) und (LF) zwei Zeichen sind.

Kommando K (Kill)

Das K-Kommando löscht Zeilen aus dem Speicherpuffer und besitzt die Formen:

```
nK, -nK
```

wobei n die Anzahl der zu löschenden Zeilen festlegt. Ein positives n löscht n Zeilen nach dem CP. Ein negatives n löscht n Zeilen vor dem CP. Wenn keine Zahl n spezifiziert ist, löscht ED die aktuelle Zeile. Befindet sich der CP am Anfang der zweiten Zeile im folgenden Beispiel:

```
Emily Dickinson said,(CR)(LF)
^"I find ecstasy in living -(CR)(LF)
the mere sense of living(CR)(LF)
is joy enough."(CR)(LF)
```

löscht das Kommando -K die vorhergehende Zeile aus dem Puffer:

```
^"I find ecstasy in living -(CR)(LF)
the mere sense of living(CR)(LF)
is joy enough."(CR)(LF)
```

Befindet sich der CP in der Mitte einer Zeile, löscht ein K-Kommando nur die Zeichen hinter dem CP bis zum Zeilenende und verbindet die Zeichen vor dem CP mit der nächsten Zeile. Ein -K löscht alle Zeichen zwischen dem Anfang der vorhergehenden Zeile und dem CP. Ein OK löscht die Zeichen der Zeile bis zum CP. Mit dem Zeichen # kann der gesamte Text zwischen dem CP und dem Pufferende bzw. Pufferanfang (-#) gelöscht werden. Mit #K ist sorgsam umzugehen, da gelöschte Zeilen nicht so einfach wiedergewonnen werden können.

## 5.4.4. Einfügen von Zeichen in den Speicherpuffer

Kommando I (Insert)

Mit dem I-Kommando können über die Tastatur Zeichen in den Speicherpuffer eingegeben werden. Das I-Kommando hat folgende Formen:

```
I
Istring^Z
```

Durch das erste Kommando wird der Eingabemodus gesetzt. In diesem Modus gelangen alle Tastatureingaben direkt in den

Speicherpuffer. ED fügt die Zeichen in Zeilen ein und beginnt eine neue Zeile, nachdem die RETURN-Taste betätigt wurde.

```
A>ED B:QUOTE.TEX

NEW FILE
 :*i

1:Emily Dickinson said,

2:"I find ecstasy in living -

3:the mere sense of living

4:is joy enough."

5:^Z

 :*
```

Die Rückkehr aus dem Eingabemodus in den Kommandomodus erfolgt mit CTRL/Z oder ESC. Wenn die ED-Meldung * auf dem Bildschirm erscheint, befindet sich ED im Kommandomodus. Im Kommandomodus können die Steuerzeichen des SCP für die Zeileneditierung benutzt werden, die in der folgenden Tabelle aufgeführt sind.

Tabelle 12: SCP-Steuerzeichen zur Zeileneditierung

| Kommando | Ergebnis |
|---|---|
| CTRL/C | Abbruch des ED und Rückkehr zum SCP. |
| CTRL/E | Rückkehr zum Zeilenanfang bei langen Zeilen ohne Übertragung der Kommandozeile in den Puffer. |
| CTRL/H | Löschen des zuletzt auf der aktuellen Zeile geschriebenen Zeichens. |
| CTRL/U | Löschen der gesamten zuletzt geschriebenen aktuellen Zeile. |
| CTRL/X | Gleiche Wirkung wie CTRL/U. |
| Rubout | Löschen des letzten Zeichens und Echo des gelöschten Zeichens auf dem Bildschirm. |

Im Eingabemodus existieren die gleichen Steuerkommandos mit Ausnahme von CTRL/C und CTRL/E.

Die Verwendung der Umschalttaste (caps-lock) auf der Tastatur bewirkt die Erzeugung von entweder Groß- oder Kleinbuchstaben bzw. von Ziffern oder Sonderzeichen. Bei der Arbeit mit SCP wird nicht zwischen Groß- und Kleinbuchstaben unterschieden, wohl aber zwischen Ziffern und Sonderzeichen. Aus diesem Grunde muß bei der kombinierten Eingabe von Buchstaben (insbesondere Großbuchstaben) und Ziffern laufend die Umschalttaste betätigt werden. ED bietet zwei Wege zur Überführung der eingegebenen Zeichen in Großbuchstaben ohne Änderung der Ziffern und ohne Verwendung der Umschalttaste. Der erste Weg ist die Eingabe des Kommandos I als Großbuchstabe. Alle Buchstaben die während der Wirkung dieses Kommandos, entweder im Eingabemodus oder in einer Zeichenkette, eingegeben werden, werden in Großbuchstaben umgewandelt. (Wird

das Kommando i als Kleinbuchstabe benutzt, bleiben alle danach eingegebenen Zeichen unverändert, wie sie geschrieben wurden). Die zweite Methode besteht in der Eingabe des U-Kommandos vor dem Text. Die Umsetzung in Großbuchstaben bleibt solange wirksam, bis -U eingegeben wird.

Kommando Istring ^Z(Insert string)

Die zweite Form des I-Kommandos schaltet nicht in den Eingabemodus um. Die Zeichenkette (string) wird in den Speicherpuffer eingefügt. ED meldet sich wieder mit *. Dabei kann die Kommandozeile mit den in Tabelle 12 zusammengestellten Steuerzeichen editiert werden.
Vor dem Einfügen einer Zeichenkette muß der CP auf die Position gebracht werden, wo der Text eingefügt werden soll. Wenn z.B. eine Zeichenfolge vor den Anfang der ersten Zeile eingefügt werden soll, wird der CP mit dem B-Kommando an den Pufferanfang positioniert und wie folgt die Zeichenkette angegeben:

```
iIn 1870, ^Z
```

Damit wird die Zeichenkette In 1870, vor dem Zeilenanfang eingefügt, danach erfolgt die ED-Meldung *. Im Speicherpuffer steht der CP nach der eingefügten Zeichenfolge:

```
In 1870, ^Emily Dickinson said,(CR)(LF)
```

## 5.4.5. Ersetzen von Zeichen

Kommando S (Substitute)

Das S-Kommando sucht im Speicherpuffer nach der spezifizierten Zeichenfolge und ersetzt sie. Das S-Kommando besitzt folgende Form:

```
nSsearch string^Znew string{^Z}
```

wobei n die Anzahl der durchzuführenden Substitutionen ist. Ist kein n spezifiziert, sucht ED nach der nächsten Übereinstimmung der spezifizierten Zeichenfolge mit dem Text im Speicherpuffer. Z.B. sucht das Kommando:

```
sEmily Dickinson^ZThe poet
```

nach der ersten Übereinstimmung von Emily Dickinson und substituiert dafür The poet. Im Speicherpuffer steht dann:

```
The poet ^ said,(CR)(LF)
```

Wurde die Umsetzung in Großbuchstaben mit dem S-Kommando wirksam, sucht ED nach der Zeichenfolge in Großbuchstaben und fügt auch Großbuchstaben bei der Substitution ein.

Wenn das S-Kommando mit anderen Kommandos kombiniert wird, muß new string mit CTRL/Z beendet werden.

## 5.5. ED-Kombinationskommandos

Durch Kombinieren von Editier- und Anzeigekommandos können Tastatureingaben und Editierzeit eingespart werden. Dazu wird eine beliebige Zahl von ED-Kommandos auf die gleiche Zeile geschrieben. ED führt diese Kommandos erst nach Eingabe von RETURN aus. Mit den SCP-Steuerkommandos zum Editieren können diese ED-Kommandoketten manipuliert werden. Wenn mehrere ED-Kommandos auf einer Zeile stehen, führt sie ED in der gleichen Reihenfolge wie sie eingegeben wurden, von links nach rechts, aus. Es gibt vier Vorschriften für die Kombination von ED-Kommandos:

- Die Zeile mit kombinierten Kommandos darf maximal 128 Zeichen enthalten.

- Wenn die kombinierte Kommandofolge eine Zeichenkette enthält, darf die Zeile 100 Zeichen nicht überschreiten.

- Kommandos zur Beendigung eines Editiervorganges dürfen nicht mit in eine Kombination einbezogen werden.

- Kommandos wie z.B. I, S, J, X und R, die Zeichenketten oder Dateispezifikationen fordern, müssen entweder als letztes Kommando einer Kombination stehen oder mit CTRL/Z bzw. ESC begrenzt werden, auch dann, wenn keine Zeichenkette oder Dateispezifikation angegeben wird.

Während die Beispiele im vorangegangenen Abschnitt den Speicherpufferinhalt und die Position des CP zeigten, wird bei den Beispielen in diesem Abschnitt gezeigt, wie sich der Bildschirminhalt während eines Editiervorganges verändert. Es sei hier nochmals daran erinnert, daß der CP imaginär ist und der Anwender sich seine Position vorstellen und merken muß, weil die ED-Kommandos Text in Relation zum CP editieren und anzeigen.

### 5.5.1. Positionieren des Zeichenzeigers

Das Positionieren des CP an das Ende einer Zeile ohne Berechnung der Zeichenanzahl wird durch Kombination eines L-Kommandos mit einem C-Kommando erreicht, L-2C. In dieser Kommandofolge werden die Zeichen (CR)(LF) als das Zeilenende erkannt.
Durch Verändern des C-Kommandos in dieser Kommandofolge, kann der CP weiter nach links positioniert werden. Diese Kommandokombination wird benutzt, wenn am Zeilenende ohne Auszählen der Zeichenanzahl editiert werden soll.

```
1:*T
  -
1: Emily Dickinson said,
1:*L-7CT
   ------
said,
1:*
```

### 5.5.2. Textanzeige

Ein T-Kommando zeigt Text ab CP bis Zeilenende an. Zur Anzeige der ganzen Zeile werden ein L- und ein T-Kommando kombiniert. Mit OLT wird der CP an den Anfang der Zeile bewegt und danach die gesamte Zeile angezeigt. Im nächsten Beispiel befindet sich der CP in der Zeilenmitte. OL bewegt den CP an den Beginn der Zeile. T zeigt ab CP bis Zeilenende den Text an, in diesem Fall die ganze Zeile:

```
3:*T
   -
sense of living
3:*0LT
   ---
3: the mere sense of living
3:*
```

Das Kommando OTT zeigt die ganze Zeile ohne Positionierung des CP an.

Um zu prüfen, ob ein ED-Kommando den CP richtig positioniert hat, wird das Kommando mit dem T-Kommando kombiniert, das die Zeile anzeigt. Im folgenden Beispiel wird ein C-Kommando mit einem T-Kommando kombiniert:

```
2:*8CT
   ---
ecstasy in living -
2:*

4:*B#T
   ---
1: Emily Dickinson said,
2: "I find ecstasy in living -
3: the mere sense of living
4: is joy enough."
1:*
```

### 5.5.3. Editieren

Um Text zu editieren und die Korrektur schnell zu prüfen, werden Editierkommandos mit weiteren ED-Kommandos kombiniert, die den CP bewegen und Text anzeigen. Kommandofolgen, wie im nächsten Beispiel, bewegen den CP, löschen spezifizierte Zeichen und prüfen die Korrektur:

```
1:*15C5D0LT
   -------

1: Emily Dickinson,
1:*
```

Wird das K-Kommando mit anderen ED-Kommandos kombiniert, so können ganze Zeilen gelöscht und die Korrektur überprüft werden:

```
1:*2L2KB#T
   --------
   1: Emily Dickinson said,
   2: "I find ecstasy in living -
   1:*
```

Die zweite Form des I-Kommandos erlaubt eine einfache Textänderung. Um diesen Editiervorgang zu prüfen, wird die I-Kommandokette mit dem C-Kommando und der OLT-Kommandofolge kombiniert. Dabei sei daran erinnert, daß die Zeichenkette des I-Kommandos mit CTRL/Z zu beenden ist:

```
1:*2OCi to a friend^ZOLT
   --------------------
1: Emily Dickinson said to a friend,
1:*
```

## 5.6. Fortgeschrittene ED-Kommandos

Die bisher beschriebenen Basiskommandos zum Editieren führen alle Editieraufgaben aus. Die folgenden Kommandos vergrößern die Effektivität des ED.

### 5.6.1. Positionieren des Zeichenzeigers und Textanzeige

#### Kommando P (Page)

Obwohl mit dem T-Kommando beliebige Texte auf dem Bildschirm angezeigt werden können, ist es manchmal angebrachter den Textpuffer mit dem P-Kommando abzuarbeiten, um damit komplette bildschirmfüllende Anzeigen von Daten zu bewirken und den CP gleichzeitig an den Anfang jeder Bildschirmseite zu bewegen. Das P-Kommando hat die Formen:

```
nP, -nP
```

wobei n die Zahl der Seiten angibt, die anzuzeigen sind. Wenn keine Seitenzahl angegeben wird, wird der CP um 23 Zeichen vorwärts bewegt und die weiteren 23 folgenden Zeilen auf dem Bildschirm ausgegeben. Damit zeigt der CP auf das erste Zeichen des Bildschirmes.

Mit OP wird die aktuelle Seite angezeigt, ohne daß dabei der CP bewegt wird. Die Angabe von 0 verhindert die CP-Bewegung. Bei negativer Seitenzahl zeigt P die Seiten in Richtung Dateianfang an.

#### Kommando n: (Line Number)

Wenn eine Zeilennummer angezeigt wird, akzeptiert ED eine Zeilennummer im Kommando, um das Ziel für den CP zu spezifizieren. Die Form des Zeilennummerkommandos ist:

```
n:
```

wobei n: die Nummer der Zielzeile ist. Dieses Kommando plaziert den CP an den Anfang der spezifizierten Zeile. Das Kommando 4: bewegt den CP z.B. an den Anfang der vierten Zeile. Es sei daran erinnert, daß ED die Textzeilen dynamisch umnumeriert, wenn im Puffer eine Zeile eingefügt oder gelöscht wurde. Deshalb kann sich während des Editierens die Nummer der Zielzeile ändern.

Kommando :n (Through Line Number)

Das inverse Kommando zum Zeilennummerkommando wird bis zu einer bestimmten Zeilennummer ausgeführt. Dieses Kommando kann nur in Verbindung mit drei ED-Kommandos benutzt werden, dem T-Kommando (type), dem L-Kommando (line), und dem K-Kommando (kill).
Das :n-Kommando hat folgende Form:

```
:ncommand
```

wobei n die Nummer der Zielzeile ist, bis zu der das Kommando angewendet werden soll.

Der Teil :n des Kommandos bewegt den CP nicht, das geschieht durch das folgende Kommando. n: kann mit :n kombiniert werden, um einen Bereich von Zeilen zu spezifizieren, in dem das folgende Kommando ausgeführt werden soll. Z.B. wird durch das Kommando 2::4T die zweite, dritte und vierte Zeile wie folgt angezeigt:

```
1:*2::4T
  -----
2: "I find ecstasy in living -
3: the mere sense of living
4: is joy enough."
2:*
```

## 5.6.2. Suchen und Ersetzen von Zeichenketten

ED unterstützt ein Suchkommando F, das den Speicherpuffer nach einem oder mehreren Worten durchsucht und den CP hinter der gefundenen Zeichenkette positioniert. Mit dem N-Kommando wird anstatt nur des Speicherpuffers die ganze Quelldatei ab der aktuellen CP-Position durchsucht. Das J-Kommando sucht nach einer Zeichenkette, fügt eine Zeichenkette ein und löscht bis zu einer dritten Zeichenkette.

Kommando F (Find)

Das F-Kommando führt die einfachsten Suchfunktionen aus. Seine Form ist:

```
nFstring{^Z}
```

wobei n eine Zahl ist, die das n-te Auftreten der gesuchten Zeichenfolge spezifiziert. Die eingegebene Zahl n muß positiv sein, da ED im Puffer ab dem CP nur vorwärts suchen kann (in Richtung Pufferende). Wird für n keine Zahl angegeben, sucht ED nach dem nächsten Auftreten der Zeichenfolge in der Datei. Im folgenden Beispiel wird nach dem zweiten Auftreten des Wortes living gesucht:

```
1:*2fliving
  --------
3:*
```

Der CP wird auf die dritte Zeile hinter die gesuchte Zeichenkette positioniert, wo das zweite Auftreten des Wortes living lokalisiert wurde. Um diese Zeile anzuzeigen, wird das F-Kommando mit einem T-Kommando kombiniert. Dabei ist zu beachten, daß das F-Kommando nach der Zeichenkette mit CTRL/Z beendet werden muß,

wenn auf der gleichen Zeile ein anderes ED-Kommando folgt:

```
1:*2fliving^Z0lt
  ------------
3:*the mere sense of living
```

Es ist ein Unterschied, ob das F-Kommando mit Groß- oder Kleinbuchstaben angegeben wird. Wenn F eingegeben wird, wandelt ED intern die Zeichenkette in Großbuchstaben um. Mit einem f-Kommando dagegen bleibt die Zeichenkette unverändert. Z.B. sucht FScp 1700 nach SCP 1700, aber fScp 1700 nach Scp 1700. Wenn ED die Zeichenkette im Speicherpuffer nicht findet, erfolgt die Mitteilung:

```
BREAK "#"AT
```

wobei das Symbol # angibt, daß der Fehler während der Ausführung eines F-Kommandos auftrat.

Kommando N

Das N-Kommando dehnt die Suche über den Speicherpuffer hinaus auf die gesamte Quelle aus. Wenn die Suche erfolgreich war, steht der CP nach der gefundenen Zeichenkette. Die Form des N-Kommandos ist:

```
nNstring{^Z}
```

wobei n eine Zahl ist, die das n-te Auftreten der gesuchten Zeichenkette bedeutet. Wird keine Zahl angegeben, sucht ED nach der nächsten Übereinstimmung der Zeichenketten. In dieser Hinsicht ist die Wirkung von N- und F-Kommando gleich. Der Zeichenkette muß ein CTRL/Z folgen, wenn noch ein weiteres ED-Kommando angehängt werden soll. Wenn ein N-Kommando ausgeführt wird, durchsucht ED den Speicherpuffer nach der Zeichenkette, wenn diese nicht gefunden wird, erfolgt keine Fehlermitteilung. Statt dessen schreibt ED automatisch die durchsuchten Daten vom Puffer in die neue Datei. Danach führt ED das Kommando 0A aus, um den Puffer mit noch nicht durchsuchten Daten aus der Quelldatei zu füllen. ED setzt die Suche im Puffer fort und schreibt die Daten an das aktuelle Ende der neuen Datei, bis entweder die Zeichenkette gefunden oder das Ende der Quelldatei erreicht wird. Wird die Zeichenkette bis zum Ende der Datei nicht gefunden, so erfolgt die Mitteilung:

```
BREAK"#"AT
```

Da ED die durchsuchten Daten in die neue Datei schreibt, bevor weitere Daten aus der Quelldatei durchsucht werden, schreibt ED gewöhnlich die Pufferinhalte zur neuen Datei, bevor das Ende der Quelldatei gefunden und die Fehlermitteilung ausgegeben wird.

Wenn ein Editiervorgang fortgesetzt werden soll, nachdem die Quelldatei abgearbeitet und der Speicherpuffer leer ist, muß das H-Kommando angewendet werden.

Kommando J (Juxtapose)

Das J-Kommando fügt eine Zeichenkette nach einer zu suchenden Zeichenkette ein, dann löscht es alle Zeichen zwischen dem Ende

der eingefügten Zeichenkette und dem Anfang einer dritten Zeichenkette. Die Form des J-Kommandos ist:

```
nJsearch string^Zinsert string^Zdelete-to string{^Z}
```

wobei n angibt, wie oft das Kommando ausgeführt wird. Zuerst wird search string im Speicherpuffer gesucht. Im folgenden Beispiel sucht ED nach Dickinson und fügt told to friend ein, danach werden alle Zeichen bis zum Komma gelöscht:

```
1:*#T
   --
1: Emily Dickinson said,
2: "I find ecstasy in living -
3: the mere sense of living
4: is joy enough."
1:*jDickinson^Z told a friend^Z,
   ----------------------------
1:*01t
   ---
1: Emily Dickinson told a friend,
1:*
```

Wenn dieses Kommando mit einem anderen kombiniert wird, muß die Zeichenkette bis zu der gelöscht werden soll, mit CTRL/Z oder ESC begrenzt werden, wie im nächsten Beispiel gezeigt wird. Wenn J als Großbuchstabe spezifiziert ist, sucht ED die spezifizierten Zeichenketten ebenfalls in Großbuchstaben und fügt auch Zeichenketten in Großbuchstaben ein.

Das J-Kommando läßt sich besonders gut auf Kommentare in Assemblerquellprogrammen anwenden:

```
236: SORT    LXI    H,SW    ;ADDRESS TOGGLE SWITCH
236:*j;^ZADDRESS SWITCH TOGGLE^Z^L^ZOLT
      ---------------------------------
236: SORT    LXI    H,SW    ;ADDRESS SWITCH TOGGLE
236:*
```

In diesem Beispiel sucht ED nach dem ersten Semikolon und fügt ADDRESS SWITCH TOGGLE nach der Marke ein, danach löscht es alle Zeichen bis zur Folge (CR)(LF), die durch CRTL/L dargestellt ist. In jeder gesuchten Zeichenkette kann CTRL/L zur Darstellung von (CR)(LF) benutzt werden, wenn sie sich über das Zeilenende ausdehnt. Ebenso kann CTRL/I in den gesuchten Zeichenketten den Tabulator darstellen.

Ist die Kommandozeile länger als eine Bildschirmzeile, wird durch Eingabe von CTRL/E ein physisches RETURN auf dem Bildschirm bewirkt. Die Kommandozeile wird dabei nicht zum ED gesendet. Eine ED-Kommandozeile mit Zeichenkette darf 100 Zeichen nicht überschreiten. Ein Kommando wird durch Betätigen der RETURN-Taste beendet. Danach wird dieses Kommando zum ED gesendet.

## Kommando M (Macro)

Ein M-Kommando kann ein Zeichenkettenkommando noch effektiver machen. Mit dem M-Kommando werden ED-Kommandos, die wiederholt ausgeführt werden sollen, in Gruppen zusammengefaßt. Die Form des M-Kommandos ist:

```
nMcommand string
```

wobei n die Anzahl der Wiederholungen der Kommandokette ist. Ein negatives n ist für das M-Kommando nicht zulässig. Wird kein n spezifiziert, verwendet ED das Zeichen # und wiederholt die Kommandokette bis zum Pufferende oder zum Ende der Quelldatei in Abhängigkeit von den in der Kette enthaltenen Kommandos.
Im folgenden Beispiel werden vier Kommandos wiederholt ausgeführt, bis das Ende des Speicherpuffers erreicht ist:

```
1:*mfliving^Z-6diLiving^Z0lt
   ------------------------
   2: "I find ecstasy in Living -
   3: the mere sense of Living
BREAK"#"AT^Z
3:
```

Der Begrenzer für ein M-Kommando ist RETURN, deshalb muß ein M-Kommando das letzte in einer Kommandozeile sein. Außerdem müssen alle Zeichenketten, die in einem Makro angewendet werden, durch CRTL/Z oder ESC begrenzt werden. Wenn eine Zeichenkette ein kombiniertes Kommando beendet, dann muß diese mit CTRL/Z abgeschlossen werden, und danach ist das M-Kommando mit RETURN zu beenden. Die Ausführung eines Makrokommandos endet immer mit der Ausschrift BREAK"#", auch, wenn eine Wiederholungszahl angegeben wurde und ED nicht das Ende des Puffers oder der Quelldatei erreicht hat. Normalerweise ist der in der Ausschrift angezeigte Kommandobuchstabe ein Kommando aus der Kommandokette und nicht M.

Ein Makrokommando wird durch Eingabe von CTRL/C abgebrochen.

### 5.6.3. Transport von Textblöcken

Um eine Gruppe von Zeilen aus einem Datenbereich in einen anderen zu transportieren, wird zunächst ein X-Kommando benutzt, das den Textblock in eine temporäre .LIB-Datei schreibt, danach werden die Originalzeilen mit einem K-Kommando gelöscht und schließlich wird der Block mit einem R-Kommando auf den Zielbereich geschrieben.

#### Kommando X (Xfer)

Das X-Kommando hat die Formen:

```
nX{^Z}
nX filespec{^Z}
```

wobei n eine Zahl von Zeichen ab CP in Richtung Pufferende definiert, die in die temporäre Datei zu übertragen sind. Deshalb muß n immer ein positiver Wert sein. Wird kein Dateiname angegeben, nimmt ED X␣␣␣␣␣␣␣␣ an. Ist kein Dateityp angegeben, wird .LIB zugewiesen. Steht das X-Kommando nicht am Ende einer Zeile, muß es mit CTRL/Z oder ESC begrenzt werden. Im folgenden Beispiel wird eine Zeile in eine temporäre Datei übertragen:

```
1:*X

1:*t
```

```
1:*Emily Dickinson said,
1:*kt

1:*"I find ecstasy in living -
1:*
```

Ist keine temporäre Datei spezifiziert, sucht ED nach einer Datei Xuuuuuuuu.LIB. Existiert diese Datei nicht, wird sie von ED erstellt. Wurde durch ein vorhergehendes X-Kommando bereits die temporäre Datei erzeugt, dann hängt ED die spezifizierten Zeilen an das aktuelle Ende dieser existierenden Datei. Mit dem Kommando OX können derartige Dateien wieder gelöscht werden. Die temporäre Datei Xuuuuuuuu.LIB wird bei Beendigung des Editierens mit einem E-, H-, O- oder Q-Kommando von ED automatisch gelöscht.

Kommando R (Read)

Das X-Kommando liest n Zeilen ab der aktuellen Zeile in eine temporäre Datei. Mit dem R-Kommando können diese Zeilen zurückübertragen werden. Das R-Kommando besitzt die Form:

```
R{^Z}
Rfilespec{^Z}
```

Ist kein Dateiname angegeben, wird Xuuuuuuuu angenommen. Ein nicht angegebener Dateityp wird durch .LIB ersetzt. R fügt die temporäre Datei vor dem CP ein. Deshalb zeigt der CP nach dem Einfügen der Datei .LIB im Speicherpuffer auf das gleiche Zeichen wie vor dem Lesen, obwohl dieses Zeichen auf einer anderen Zeile steht. Soll das R-Kommando mit anderen Kommandos kombiniert werden, muß der Dateiname vom nachfolgenden Kommando mit CTRL/Z getrennt werden, wie im folgenden Beispiel, in dem ED eine eingelesene Datei auf dem Bildschirm wiedergibt:

```
1:*4l

 :*R^ZB#T

1: "I find ecstasy in living -
2: the mere sense of living
3: is joy enough."
4: Emily Dickinson said,
1:*
```

## 5.6.4. Dateisicherung und ED-Ende

Mit den folgenden drei Kommandos können editierte Daten gerettet und das Editieren beendet werden.

Kommando H (Head of File)

Ein H-Kommando sichert den Inhalt des Speicherpuffers, ohne ED abzubrechen und kehrt zum Kopf (head) der Datei zurück. Auf die bereits editierten Daten kann erneut zugegriffen werden, ohne ED abzubrechen. Das H-Kommando besitzt die Form:

```
H
```

gefolgt von RETURN.

Bei der Ausführung eines H-Kommandos vollendet ED zuerst die neue Datei durch Übertragung aller restlichen Zeilen aus dem Puffer und der Quelldatei in die neue Datei. Dann wird die neue Datei geschlossen, die BAK-Datei mit dem gleichen Namen wie die Quelldatei gelöscht und die Originaldatei in den Typ BAK umbenannt. ED weist dann der neuen Datei, die bisher den Dateityp .¤¤¤ besaß, den Namen der Orginaldatei zu. Schließlich eröffnet ED die neue umbenannte Datei wie eine Quelldatei zum erneuten Editieren. Gleichzeitig wird eine neue Datei mit dem Typ .¤¤¤ eröffnet. Wenn sich ED mit * meldet, steht der CP am Anfang eines leeren Speicherpuffers. Sollen die editierten Daten zu einer anderen als der Originaldatei gesendet werden, wird das folgende Kommando benutzt:

```
A>ED filespec differentfilespec
```

Wenn danach mit dem H-Kommando ein Restart des ED durchgeführt und das Editieren mit einem E-Kommando beendet wird, wird die Datei differentfilename.¤¤¤ in differentfilename.BAK umbenannt und eine neue Datei differentfilespec erstellt.

Kommando O (Original)

Ein O-Kommando schließt den Editiervorgang ab und erlaubt die Rückkehr zur Originaldatei zum erneuten Editieren, ohne ED abzubrechen. Das O-Kommando besitzt die Form:

```
O
```

gefolgt von RETURN.

Bei der Eingabe des O-Kommandos wird sicherheitshalber abgefragt:

```
O(Y/N)?
```

Geantwortet wird mit Y oder N. Bei Betätigung einer beliebigen anderen Taste wird die Frage wiederholt. Nach Y löscht ED die temporäre Datei und den Inhalt des Speicherpuffers. Wenn die ED-Meldung * erscheint, zeigt der CP auf den Anfang eines leeren Speicherpuffers genau wie beim Neustart des ED.

Kommando Q (Quit)

Ein Q-Kommando schließt den Editiervorgang ab und beendet ED. Das Q-Kommando hat die Form:

```
Q
```

gefolgt von RETURN.

Bei der Eingabe des Q-Kommandos wird sicherheitshalber abgefragt:

```
Q(Y/N)?
```

Geantwortet wird mit Y oder N. Jedes andere Zeichen führt zur Wiederholung der Frage.

Wird mit Y geantwortet, löscht ED die temporäre Datei, schließt die Quelldatei und gibt die Steuerung an das SCP zurück.

Auch mit CTRL/C kann die Steuerung an das SCP zurückgegeben werden. Damit wird aber verhindert, daß ED die Quelldatei oder die neue Datei schließen und temporäre Dateien löschen kann.

## 5.7. ED-Fehlermitteilungen

Beim ED gibt es zwei Typen von Fehlermitteilungen. Entweder kann ED ein Editierkommando nicht ausführen oder er kann infolge einer SCP-Fehlermitteilung die spezifizierte Datei nicht lesen oder schreiben.

Die Form der ED-Fehlermitteilung ist:

```
BREAK "x" AT c
```

wobei x eines der in der folgenden Tabelle definierten Symbole ist und c der Buchstabe des Kommandos, bei dem der Fehler auftritt.

Tabelle 12: ED-Fehlersymbole

| Symbol | Bedeutung |
|---|---|
| # | Suchfehler. Die in einem F, S oder N-Kommando angegebene Zeichenkette kann nicht gefunden werden. |
| ?c | Unbekannter Kommandobuchstabe . ED kennt das Kommando nicht oder ein E-, H-, oder O-Kommando befindet sich nicht allein auf einer Kommandozeile. |
| O | Keine .LIB-Datei. ED findet die im R-Kommando spezifizierte LIB-Datei nicht. |
| > | Puffer voll. ED kann keine Zeichen mehr in den Puffer eintragen oder die spezifizierte Zeichenkette in einem F-, N- oder S-Kommando ist zu lang. |
| E | Kommandoabbruch. Eine Tastatureingabe hat die Ausführung des Kommandos abgebrochen. |
| F | Dateifehler. Es folgt entweder DISK FULL oder DIRECTORY FULL. |

Die folgenden Beispiele zeigen, wie auf ED-Fehlermitteilungen reagiert werden kann:

```
BREAK ">" AT A
```

bedeutet, daß ED den Speicherpuffer bereits gefüllt hat, ehe ein A-Kommando vollständig ausgeführt wurde. Das bedeutet, daß sich der CP am Pufferende befindet und kein Editieren möglich ist. Reagiert wird entweder mit dem 0w-Kommando, um im Puffer Platz zu schaffen oder mit einem O- bzw. H-Kommando zum Restart des ED.

```
BREAK "#" AT F
```

bedeutet, daß ED das Pufferende erreicht hat, ohne die im F-Kommando spezifizierte Zeichenkette zu finden. Dabei befindet sich der CP am Pufferende. Der CP muß mit einem B- oder n:-Kommando positioniert werden, um das Editieren fortsetzen zu können.

```
BREAK "F" AT F
DISK FULL
```

Unnötige Dateien auf der Platte müssen mit dem OX-Kommando gelöscht werden oder der Pufferinhalt muß mit B#Xd:buffer.sav auf eine andere Platte gerettet werden.

```
BREAK "F" AT n
DIRECTORY FULL
```

Darauf wird mit den gleichen Kommandos reagiert, die im vorhergehenden Beispiel angegeben wurden.

Die folgende Tabelle enthält die ED-Mitteilungen, die bei einem Plattendateifehler erscheinen, wenn die Datei nicht gelesen oder geschrieben werden kann.

Tabelle 13: ED-Mitteilungen zu Plattendateifehlern

| Mitteilung | Bedeutung |
|---|---|
| BDOS ERR ON d: RO | Die Platte d: besitzt ein RO-Attribut. Das tritt ein, wenn die Platte nach dem letzten Start des SCP gewechselt wurde. |
| ** FILE IS READ ONLY ** | Die im Startkommando spezifizierte Datei besitzt ein RO-Attribut. ED kann die Datei nur lesen, so daß sie zwar geprüft, aber nicht verändert werden kann. |

## Anlage 1 KOI7- und Hexadezimalkonvertierung

Der KOI7-Code ist festgelegt in der Codetabelle KOI7H0 im RGW-Standard ST RGW 356-76. Bei Bezugnahme auf diese Tabelle wird die Kurzform KOI7 verwendet.

Tabelle 14 definiert die KOI7-Symbole, die Tabelle 15 zeigt eine Liste der KOI7- und Hexadezimalkonvertierung. Die Tabelle enthält binäre, dezimale, hexadezimale und KOI7-Konvertierungen.

Tabelle 14: KOI7-Symbole

| Symbol | Bedeutung |
|---|---|
| ACK | Positive Rückmeldung (acknowledge) |
| BEL | Klingel (bell) |
| BS | Rückwärtsschritt (backspace) |
| CAN | Ungültig (cancel) |
| CR | Wagenrücklauf (carriage return) |
| DC | Gerätesteuerung (device control) |
| DEL | Löschen (delete) |
| DLE | Datenübertragungsumschaltung (data link escape) |
| EM | Ende der Aufzeichnung (end of medium) |
| ENQ | Stationsaufforderung (enquiry) |
| EOT | Ende der Übertragung (end of transmission) |
| ESC | Umschaltung (escape) |
| ETB | Ende des Datenübertragungsblockes (end of transmission block) |
| ETX | Ende des Textes (end of text) |
| FF | Fomularvorschub (form-feed) |
| FS | Hauptgruppentrennung (file separator) |

Tabelle 14: KOI7-Symbole

| Symbol | Bedeutung |
|---|---|
| GS | Gruppentrennung (group separator) |
| HT | Horizontaltabulator (horizontal tabulation) |
| LF | Zeilenvorschub (line-feed) |
| NAK | Negative Rückmeldung (negative acknowledge) |
| NUL | Null (null) |
| RS | Untergruppentrennung (record separator) |
| SI | Rückschaltung (shift in) |
| SO | Dauerumschaltung (shift out) |
| SOH | Anfang des Kopfes (start of heading) |
| SP | Zwischenraum/Leerschritt (space) |
| STX | Anfang des Textes (start of text) |
| SUB | Substitution (substitute) |
| SYN | Synchronisation (synchronous idle) |
| US | Teilgruppentrennung (unit separator) |
| VT | Vertikaltabulator (vertical tabulation) |

Tabelle 15: KOI7-Konvertierungstabelle

| Binär | Dezimal | Hexadezimal | KOI7 |
|---|---|---|---|
| 0000000 | 0 | 0 | NUL |
| 0000001 | 1 | 1 | SOH (CTRL/A) |
| 0000010 | 2 | 2 | STX (CTRL/B) |
| 0000011 | 3 | 3 | ETX (CTRL/C) |
| 0000100 | 4 | 4 | EOT (CTRL/D) |
| 0000101 | 5 | 5 | ENQ (CTRL/E) |
| 0000110 | 6 | 6 | ACK (CTRL/F) |
| 0000111 | 7 | 7 | BEL (CTRL/G) |
| 0001000 | 8 | 8 | BS (CTRL/H) |
| 0001001 | 9 | 9 | HT (CTRL/I) |
| 0001010 | 10 | A | LF (CTRL/J) |
| 0001011 | 11 | B | VT (CTRL/K) |
| 0001100 | 12 | C | FF (CTRL/L) |
| 0001101 | 13 | D | CR (CTRL/M) |
| 0001110 | 14 | E | SO (CTRL/N) |
| 0001111 | 15 | F | SI (CTRL/O) |
| 0010000 | 16 | 10 | DLE (CTRL/P) |
| 0010001 | 17 | 11 | DC1 (CTRL/Q) |
| 0010010 | 18 | 12 | DC2 (CTRL/R) |
| 0010011 | 19 | 13 | DC3 (CTRL/S) |
| 0010100 | 20 | 14 | DC4 (CTRL/T) |
| 0010101 | 21 | 15 | NAK (CTRL/U) |
| 0010110 | 22 | 16 | SYN (CTRL/V) |
| 0010111 | 23 | 17 | ETB (CTRL/W) |
| 0011000 | 24 | 18 | CAN (CTRL/X) |
| 0011001 | 25 | 19 | EM (CTRL/Y) |
| 0011010 | 26 | 1A | SUB (CTRL/Z) |
| 0011011 | 27 | 1B | ESC (CTRL/[) |
| 0011100 | 28 | 1C | FS (CTRL/\) |
| 0011101 | 29 | 1D | GS (CTRL/]) |
| 0011110 | 30 | 1E | RS (CTRL/^) |
| 0011111 | 31 | 1F | US (CTRL/_) |
| 0100000 | 32 | 20 | (SPACE) |
| 0100001 | 33 | 21 | ! |
| 0100010 | 34 | 22 | " |
| 0100011 | 35 | 23 | # |
| 0100100 | 36 | 24 | ¤ |
| 0100101 | 37 | 25 | % |
| 0100110 | 38 | 26 | & |
| 0100111 | 39 | 27 | ' |
| 0101000 | 40 | 28 | ( |
| 0101001 | 41 | 29 | ) |
| 0101010 | 42 | 2A | * |
| 0101011 | 43 | 2B | + |
| 0101100 | 44 | 2C | , |
| 0101101 | 45 | 2D | - |
| 0101110 | 46 | 2E | . |
| 0101111 | 47 | 2F | / |

Tabelle 15: (Fortsetzung)

| Binär | Dezimal | Hexadezimal | KOI7 |
|---|---|---|---|
| 0110000 | 48 | 30 | 0 |
| 0110001 | 49 | 31 | 1 |
| 0110010 | 50 | 32 | 2 |
| 0110011 | 51 | 33 | 3 |
| 0110100 | 52 | 34 | 4 |
| 0110101 | 53 | 35 | 5 |
| 0110110 | 54 | 36 | 6 |
| 0110111 | 55 | 37 | 7 |
| 0111000 | 56 | 38 | 8 |
| 0111001 | 57 | 39 | 9 |
| 0111010 | 58 | 3A | : |
| 0111011 | 59 | 3B | ; |
| 0111100 | 60 | 3C | < |
| 0111101 | 61 | 3D | = |
| 0111110 | 62 | 3E | > |
| 0111111 | 63 | 3F | ? |
| 1000000 | 64 | 40 | @ |
| 1000001 | 65 | 41 | A |
| 1000010 | 66 | 42 | B |
| 1000011 | 67 | 43 | C |
| 1000100 | 68 | 44 | D |
| 1000101 | 69 | 45 | E |
| 1000110 | 70 | 46 | F |
| 1000111 | 71 | 47 | G |
| 1001000 | 72 | 48 | H |
| 1001001 | 73 | 49 | I |
| 1001010 | 74 | 4A | J |
| 1001011 | 75 | 4B | K |
| 1001100 | 76 | 4C | L |
| 1001101 | 77 | 4D | M |
| 1001110 | 78 | 4E | N |
| 1001111 | 79 | 4F | O |
| 1010000 | 80 | 50 | P |
| 1010001 | 81 | 51 | Q |
| 1010010 | 82 | 52 | R |
| 1010011 | 83 | 53 | S |
| 1010100 | 84 | 54 | T |
| 1010101 | 85 | 55 | U |
| 1010110 | 86 | 56 | V |
| 1010111 | 87 | 57 | W |
| 1011000 | 88 | 58 | X |
| 1011001 | 89 | 59 | Y |
| 1011010 | 90 | 5A | Z |
| 1011011 | 91 | 5B | [ |
| 1011100 | 92 | 5C | \ |
| 1011101 | 93 | 5D | ] |
| 1011110 | 94 | 5E | ^ |
| 1011111 | 95 | 5F | < |

Tabelle 15: (Fortsetzung)

| Binär | Dezimal | Hexadezimal | KOI7 |
|---|---|---|---|
| 1100000 | 96 | 60 | ' |
| 1100001 | 97 | 61 | a |
| 1100010 | 98 | 62 | b |
| 1100011 | 99 | 63 | c |
| 1100100 | 100 | 64 | d |
| 1100101 | 101 | 65 | e |
| 1100110 | 102 | 66 | f |
| 1100111 | 103 | 67 | g |
| 1101000 | 104 | 68 | h |
| 1101001 | 105 | 69 | i |
| 1101010 | 106 | 6A | j |
| 1101011 | 107 | 6B | k |
| 1101100 | 108 | 6C | l |
| 1101101 | 109 | 6D | m |
| 1101110 | 110 | 6E | n |
| 1101111 | 111 | 6F | o |
| 1110000 | 112 | 70 | p |
| 1110001 | 113 | 71 | q |
| 1110010 | 114 | 72 | r |
| 1110011 | 115 | 73 | s |
| 1110100 | 116 | 74 | t |
| 1110101 | 117 | 75 | u |
| 1110110 | 118 | 76 | v |
| 1110111 | 119 | 77 | w |
| 1111000 | 120 | 78 | x |
| 1111001 | 121 | 79 | y |
| 1111010 | 122 | 7A | z |
| 1111011 | 123 | 7B | { |
| 1111100 | 124 | 7C | \| |
| 1111101 | 125 | 7D | } |
| 1111110 | 126 | 7E | ~ |
| 1111111 | 127 | 7F | DEL |

## Anlage 2 SCP-1700-Dateitypen

SCP kennzeichnet jede Datei durch eine einheitliche Dateispezifikation, die aus der Laufwerkskennung, dem Dateinamen und dem Datentyp besteht. Der Dateityp ist wahlfrei und wird aus bis zu drei alphanumerischen Zeichen gebildet. Er wird vom Dateinamen durch eine Punkt getrennt. Der Dateityp zeigt im allgemeinen die spezielle Art der Datei an. Die folgende Tabelle enthält die Standard-Dateitypen und ihre Bedeutung.

Tabelle 16: Dateitypen

| Dateityp | Bedeutung |
|---|---|
| A86 | Quelle in Assemblersprache; der SCP-Assembler ASM86 übersetzt eine Datei vom Typ A86 in den Maschinencode (H86). |
| BAK | Hintergrunddatei, die vom Editor erstellt wird. Der Editor benennt die Ursprungsdatei in diesen Dateityp um, so daß zu erkennen ist, daß die Ursprungsdatei bearbeitet wurde. Als BAK-Datei kann weiterhin auf sie Bezug genommen werden. |
| CMD | Kommandodatei, die ausführbare Befehle in Maschinencode enthält. |
| COM | Datei, die auf einem 8-bit-Prozessor ausführbar ist. |
| H86 | Programmdatei im hexadezimalen Format. |
| LST | Druckbare Datei, die am Bildschirm angezeigt oder mit dem Drucker ausgegeben werden kann. |
| PRN | gleiche Bedeutung wie LST |
| SUB | Dateityp, der für SUBMIT-Eingabedateien gefordert wird, die ein oder mehrere SCP-Kommandos enthalten. Das SUBMIT-Programm führt die Kommandos der SUB-Datei dem SCP zur Stapelverarbeitung (batch mode) zu. |
| SYM | Symboltabellendatei |
| $$$ | Temporäre Datei, die von PIP erstellt wird. |

## Anlage 3 SCP-1700-Steuerzeichen

Tabelle 17: SCP-1700-Steuerzeichen

| Taste | Wirkung |
|---|---|
| BACKSPACE | bewegt den Kursor um eine Stelle zurück und löscht das vorhergehende Zeichen |
| CTRL/C | bricht das gerade laufende Programm ab |
| DEL | gleiche Wirkung wie RUB |
| CTRL/E | bewirkt ein physisches RETURN, sendet aber das Kommando nicht zum SCP |
| CTRL/H | gleiche Wirkung wie BACKSPACE |
| CTRL/J | Line-feed (Zeilenschaltung), beendet die Eingabe auf dem Terminal |
| CTRL/M | gleiche Wirkung wie RETURN |
| CTRL/P | wiederholt alle Terminalaktivitäten auf dem Drukker (Echo), ein zweites CTRL/P beendet das Echo. Das Kommando ist nur wirksam, wenn ein Drucker angeschlossen ist. |
| CTRL/R | wiederholt die aktuelle Kommandozeile. Wird sinnvoll nach RUB- oder DEL-Benutzung angewendet. |
| RETURN | Carriage return (Wagenrücklauf) |
| RUB | löscht ein Zeichen links vom Kursor, wiederholt das gelöschte Zeichen (Echo), bewegt den Kursor nach rechts |
| CTRL/S | unterbricht die Listenausgabe auf dem Teminal zeitweilig, ein erneutes CTRL/S setzt die Listenausgabe fort |
| CRTL/U | beendet die Zeile, zeigt ein # an, bewegt den Kursor auf die nächste Zeile nach unten und wartet auf ein neues Kommando |
| CRTL/X | löscht alle Zeichen in der Kommandozeile |
| CTRL/Z | Zeichenketten- oder Feldseparator |

## Anlage 4 SCP-1700-Fehlermitteilungen

Ambiguous operand

DDT86: In einem A-Kommando wurde ein mehrdeutiger Operand angegeben. Der Operand muß durch BYTE oder WORD spezifiziert werden.

Bad Directory on d:
Space Allocation Conflict:
User n d:filename.typ

STAT hat einen Speicherbelegungswiderspruch erkannt; ein Datenblock ist mehreren Dateien zugeordnet. Ein oder mehrere Dateinamen werden ausgegeben. Jede dieser Dateien enthält einen Datenblock, den schon eine andere Datei auf der Platte belegt. Dieser Fehler wird durch Löschen aller angegebenen Dateien beseitigt. Danach wird durch Eingabe von CTRL/C die Zuordnung regeneriert. Wird das unterlassen, kann sich der Fehler wiederholen.

BDOS err on d:

SCP ersetzt d: durch die Laufwerkskennung des Laufwerkes, bei dem der Fehler auftrat. Diese Ausschrift erscheint, wenn SCP keine Platte im Laufwerk findet, wenn die Platte nicht richtig formatiert ist, wenn die Platte nicht verriegelt ist oder wenn das Netz für das Laufwerk nicht eingeschaltet ist. Diese Fehlermöglichkeiten sind zu prüfen und zu beseitigen.

BDOS err on d: bad sector

Die Ursache kann ein Gerätefehler oder eine nicht richtig formatierte Platte sein. Mit CTRL/C wird das laufende Programm unterbrochen und zu SCP zurückgekehrt. Durch Eingabe von RETURN kann der Fehler ignoriert werden.

BDOS err on d: select

SCP erhielt eine Anforderung auf ein nichtexistierendes Laufwerk, oder die Platte im Laufwerk ist nicht richtig formatiert. SCP beendet das laufende Programm, sobald eine beliebige Taste betätigt wird.

BDOS err on d: RO

Dem spezifizierten Laufwerk wurde mit STAT das RO-Attribut zugewiesen, oder die Platte im Laufwerk wurde gewechselt ohne Eingabe eines CTRL/C. SCP beendet das laufende Programm, sobald eine beliebige Taste betätigt wird.

Cannot close

ASM86: Eine Ausgabedatei kann nicht geschlossen werden. Dieser Fehler beendet die Arbeit des ASM86. Der Bediener muß überprüfen, ob sich die richtige Platte im Laufwerk befindet oder ob sie

schreibgeschützt ist.
DDT86: Die Plattendatei, geschrieben mit einem W-Kommando, kann nicht geschlossen werden. Dieser Fehler beendet die Arbeit des DDT86. Der Bediener muß prüfen, ob sich die richtige Platte im Laufwerk befindet oder ob sie schreibgeschützt ist.

Command name?

Wenn SCP das eingegebene Kommando nicht finden kann, gibt es den Kommandonamen mit einem nachfolgenden Fragezeichen aus. Fehlerursachen sind ein unkorrekter Kommandoname oder das Kommando befindet sich nicht als CMD-Datei auf der Standard- bzw. der spezifizierten Platte.

DESTINATION IS R/O, DELETE (Y/N)?

PIP: Die Zieldatei, die im PIP-Kommando spezifiziert wurde, ist schreibgeschützt. Wenn Y eingegeben wird, wird diese Datei gelöscht, bevor eine Dateikopie hergestellt wird.

Directory full

ASM86: Im Verzeichnis ist kein Platz mehr für die Ausgabedateien. Entweder müssen die nicht benötigten Dateien gelöscht oder eine Platte mit freiem Speicherplatz benutzt werden.

Disk full

ASM86: Auf der Platte ist nicht mehr genug Platz für die Ausgabedateien (LST, H86, SYM). Nicht benötigte Dateien müssen gelöscht oder es muß eine Platte mit mehr freiem Speicherplatz benutzt werden.

Disk read error

ASM86: Eine Datei konnte nicht richtig gelesen werden. Gewöhnlich tritt dieser Fehler bei einem nichterkannten Dateiende (EOF) auf. Die Ursache muß in der Quelldatei beseitigt werden.
DDT86: Die im R-Kommando spezifizierte Plattendatei konnte nicht richtig gelesen werden. Ursache ist häufig ein nicht erkanntes Dateiende (EOF). Die Ursache muß in der zu lesenden Datei beseitigt werden.

Disk write error

DDT86: Während eines W-Kommandos konnte eine Schreiboperation nicht erfolgreich ausgeführt werden, weil die Platte voll ist. Nicht benötigte Dateien müssen gelöscht oder eine andere freie Platte muß benutzt werden.

Double defined variable

ASM86: Ein Variablenname ist bereits im Programm als Variablenname oder als Marke benutzt worden.
Beispiel:

```
X  DB   5
   ...
X  DB   123H
```

Double defined label

ASM86: Eine Marke ist bereits im Programm als Marke oder Variablenname benutzt worden.
Beispiel:

```
LAB3:  MOV  BX,5
       ...
LAB3:  CALL MOVE
```

Double defined symbol - treated as undefined

ASM86: Der Name einer EQU-Anweisung ist bereits im Programm als Name benutzt worden.

ERROR: BAD PARAMETER

PIP: In einem Kommando wurde ein unzulässiger Parameter eingegeben. Das Kommando ist korrekt zu wiederholen.

ERROR: CLOSE FILE -{filespec}

PIP: Eine Ausgabedatei kann nicht geschlossen werden. Der Bediener muß überprüfen, ob sich die richtige Platte im Laufwerk befindet oder ob sie schreibgeschützt ist.

ERROR: DISK READ - {filespec}

PIP: Die im Kommando spezifizierte Eingabedatei konnte nicht richtig gelesen werden. Gewöhnlich ist die Ursache eine nichterkannte Dateiendemarke (EOF). Der Fehler ist in der Datei zu beseitigen.

ERROR: DISK WRITE - {filespec}

PIP: Die Schreiboperation zur Platte konnte von PIP nicht erfolgreich ausgeführt werden, weil die Platte voll ist. Nicht benötigte Dateien sind zu löschen oder eine andere freie Platte ist zu benutzen.

ERROR: FILE NOT FOUND - {filespec}

PIP: Eine spezifizierte Eingabedatei existiert nicht.

ERROR: HEX RECORD CHECKSUM - {filespec}

PIP: Bei der Übertragung einer Hex-Datei wurde ein Prüfsummenfehler erkannt. Die Hex-Datei muß korrigiert werden, d.h. die Hex-Datei ist neu zu erstellen.

Error in codemacro building

ASM86: Entweder enthält ein Codemakro unzulässige Anweisungen oder eine Codemakroanweisung wurde außerhalb eines Codemakros erkannt.

ERROR: INVALID DESTINATION

PIP: Das in einem PIP-Kommando spezifizierte Ziel ist unzulässig. Es wurde möglicherweise ein Eingabegerät als Ziel angegeben.

ERROR: INVALID FORMAT

PIP: Das PIP-Kommando hat ein unzulässiges Format.

ERROR: INVALID HEX DIGIT - {filespec}

PIP: Während des Einlesens einer Hex-Datei wurde eine unzulässige Hex-Ziffer erkannt. Die Hex-Datei muß neu erstellt werden.

ERROR: INVALID SEPARATOR

PIP: Zwischen zwei Dateinamen wurde ein unzulässiger Separator angewendet.

ERROR: INVALID SOURCE

PIP: Die im PIP-Kommando spezifizierte Quelle ist unzulässig. Es wurde möglicherweise ein Ausgabegerät als Quelle angegeben.

ERROR: INVALID USER NUMBER

PIP: Es wurde eine Nutzernummer größer 15 angegeben. Zulässig sind Nummern von 0 bis 15.

ERROR: NO DIRECTORY SPACE - {filespec}

PIP: Im Verzeichnis ist kein Platz mehr für die Ausgabedatei. Nicht benötigte Dateien sind zu löschen oder eine andere freie Platte ist zu benutzen.

ERROR: QUIT NOT FOUND

PIP: Das Zeichenkettenargument zu einem Q-Parameter konnte in der Eingabedatei nicht gefunden werden.

ERROR: START NOT FOUND

PIP: Das Zeichenkettenargument zu einem S-Parameter konnte in der Quelldatei nicht gefunden werden.

ERROR: UNEXPECTED END OF HEX FILE - {filespec}

PIP: Es wurde ein Dateiende vor dem Ende eines Hex-Satzes erkannt. Die Hex-Datei muß neu erstellt werden.

ERROR: USER ABORTED

PIP: Vom Bediener wurde eine PIP-Operation durch Betätigen einer Taste abgebrochen.

ERROR: VERIFY - {filespec}

PIP: Wenn ein Kopiervorgang mit Vergleichslesen erfolgt und eine Differenz zwischen Kopie und Ursprungsdatei festgestellt wird, ist die Ursache für den Fehler entweder die Zielplatte oder deren Laufwerk.

File exists

Von SPC wurde verlangt, eine Datei mit bereits vorhandenem Namen zu erstellen. Entweder wird die existierende Datei gelöscht oder es ist eine andere Dateispezifikation zu benutzen.

File name syntax error

ASM86: Der Dateiname in einer INCLUDE-Anweisung ist unzulässig.
Beispiel:

```
INCLUDE FILE.A86X
```

File not found

SCP kann die spezifizierte Datei nicht finden. Es ist zu überprüfen, ob die richtige Laufwerkskennung angegeben wurde oder die richtige Platte im Laufwerk ist.

Garbage at end of line - ignored

ASM86: Trotz Zeilenende wurden weitere Zeichen erkannt.
Beispiel:

```
NOLIST 4
MOV    AX,4    RET
```

Illegal expression element

ASM86: Ein Ausdruck ist unzulässig.
Beispiel:

```
X   DB  12X
    DW  (4*)
```

Illegal first item

ASM86: Das erste Element auf der Quellzeile ist nicht zulässig als Identifikator, Mnemonik oder Direktive.
Beispiel: 1 2 3 4 H

Illegal "IF" operand - "IF" ignored

ASM86: Entweder ist der Ausdruck in einer IF-Anweisung nicht numerisch, oder er enthält eine Vorwärtsreferenz.

Illegal pseudo instruction

ASM86: Entweder fehlt ein geforderter Identifikator vor einem Pseudobefehl, oder ein Identifikator steht vor einem Pseudobefehl, der einen Identifikator nicht erlaubt.

Illegal pseudo operand

ASM86: Der Operand einer Direktive ist unzulässig.
Beispiel:

```
X   EQU  0AGH
    TITLE UNQUOTED STRING
```

Instruction not in code segment

ASM86: Ein Befehl tritt außerhalb des Codesegments auf.

Insufficient memory

DDT86: Es ist nicht ausreichend Speicherplatz vorhanden, um die in einem R- oder E-Kommando spezifizierte Datei zu laden.

Invalid Assignment

STAT: In einer STAT-Gerätezuweisung wurde ein unzulässiges Gerät angegeben. Mit dem Kommando STAT VAL: werden die zulässigen Zuweisungen angezeigt.

Label out of range

ASM86: Die Marke, auf die ein CALL, JUMP oder LOOP Bezug nimmt, befindet sich außerhalb der zulässigen Grenzen. Die Marke kann in einem anderen Segment als Befehl definiert sein. Im Falle von kurzen Befehlen (JMPS u.a.) ist die Marke mehr als 128 Bytes von der dem Befehl folgende Adresse entfernt.

Memory request denied

DDT86: Eine Speicherforderung während eines R-Kommandos konnte nicht erfüllt werden. Bis zu acht Speicherblöcke können auf einmal belegt werden (einschließlich DDT86).

Missing instruction

ASM86: Einem Präfix auf einer Quellzeile folgt kein Befehl
Beispiel: REPNZ

Missing pseudo instruction

ASM86: Das erste Element auf einer Quellzeile ist ein zulässiger Identifikator, und das zweite Element ist keine zulässige Anweisung, der ein Identifikator vorangehen darf.
Beispiel:

```
     THIS IS A MISTAKE
```

Missing segment information in operand

ASM86: Der Operand in einem CALLF- oder JMPF-Befehl (oder ein Ausdruck in einer DD-Direktive) enthält keine Segmentinformation. Die erforderliche Segmentinformation wird durch Einfügen eines numerischen Feldes in die Segmentanweisung ergänzt:

```
     CSEG   1000H
X:
       ...
     JMPF   X
     DD     X
```

Missing type information in operand(s)

ASM86: Kein Befehlsoperand enthält eine ausreichende Typinformation.
Beispiel:

```
     MOV  [BX],10
```

Nested "IF" illegal - "IF" ignored

ASM86: Die Zahl der maximal zulässigen Verschachtelungsebenen für die IF-Anweisung wurde überschritten.

Nested INCLUDE not allowed

ASM86: Die INCLUDE-Direktive erkannte eine bereits einbezogene Datei.

No file

SCP findet die spezifizierte Datei nicht, oder es existieren keine Dateien.

ASM86: Die spezifizierte Quelldatei oder einzubeziehende Datei wurde auf dem angezeigten Laufwerk nicht gefunden.
DDT86: Die in einem R- oder E-Kommando spezifizierte Datei konnte auf der Platte nicht gefunden werden.

No matching "IF" for "ENDIF"

ASM86: Es wurde eine ENDIF-Anweisung erkannt, zu der keine IF-Anweisung gehört.

No space

DDT86: Für die mit dem W-Kommando zu schreibende Datei ist kein Verzeichnisplatz frei.

Operand(s) mismatch instruction

ASM86: Entweder hat ein Befehl eine unerlaubte Zahl von Operanden oder der Typ der Operanden stimmt nicht überein.
Beispiel:

```
     MOV  CX,1,2
X    DB   0
     MOV  AX,X
```

Parameter error

ASM86: Ein Parameter im ASM86-Kommando wurde unkorrekt angegeben.
Beispiel:

```
     ASM86 TEST ¤S;
```

Symbol illegally forward referenced - neglected

ASM86: Das angezeigte Symbol hat eine unerlaubte Vorwärtsreferenz in einer ORG-, RS-, EQU- oder IF-Anweisung.

Symbol table overflow

ASM86: In der Symboltabelle ist nicht mehr genügend Speicherplatz vorhanden. Die Länge und/ oder die Zahl der Symbole ist zu verringern.

Undefined element of expression

ASM86: Ein als Operand benutzter Identifikator ist nicht definiert oder hat eine unerlaubte Vorwärtsreferenz.
Beispiel:

```
         JMP X
     A   EQU B
     B   EQU 5
         MOV  AL,B
```

Undefined instruction

ASM86: Das einer Marke auf der Quellenzeile folgende Element ist kein zulässiger Befehl.
Beispiel:
DONE: BAD INSTR

Use: [size] [ro] [rw] [sys] or [dir]

STAT: Diese Ausschrift wird von einem STAT-Kommando bei der Zuweisung eines unzulässigen Dateiattributes veranlaßt. Es dürfen nur die zulässigen Angaben benutzt werden.

Use: STAT d:=RO

STAT: Es wurde ein unzulässiges Laufwerkkommando gegeben. Nur die zulässige Laufwerkzuweisung darf benutzt werden.

Too Many Files

STAT: Ein STAT-Kommando mit Dateigruppensymbolen bezieht sich auf mehr als 512 Dateien und überschreitet damit das zulässige Maximum.

Verify error at s:o

DDT86: Der mit einem F-, S-, M- oder A-Kommando eingegebene Wert konnte nicht erfolgreich zurückgelesen werden. Ursachen sind ein Speicherdefekt, ein Schreibversuch zum ROM oder auf eine nichtexistierende Speicherzelle.

## Anlage 5 Wortverzeichnis

aktuelles Plattenlaufwerk, aktuelle Nutzernummer (default):

Einmal ausgewähltes Plattengerät bzw. Nutzernummer. Jedes Kommando, in dem nicht ein anderes Plattengerät oder eine andere Nutzernummer angegeben werden, bezieht sich auf das zuletzt angegebene Plattengerät bzw. die zuletzt angegebene Nutzernummer. Beim ersten Aufruf des SCP ist das Laufwerk A das Standardlaufwerk und 0 die Standardnutzernummer, sie gelten bis zur Veränderung mit dem Kommando USER.

Anfangsladen (bootstrap):

Prozeß des Ladens des Betriebssystems in den Speicher. Die Anfangslader variieren von System zu System. Das Anfangsladen eines Betriebssystems muß auf die vorhandene Speichergröße und die Hard-Ware-Umgebung, die das Betriebssystem bedienen soll, zugeschnitten sein. Typisch ist, daß der Anfangslader automatisch beim Einschalten des Rechners oder beim Starten (reset) aktiviert wird (sog. Kaltstart).

Anwenderprogramm (applications program):

Progamm, das für seine Ausführung ein Betriebssystem als Umgebung benötigt. Typische Anwenderprogramme sind Pakete zur Geschäfts - Buchführung, Wortverarbeitungsprogramme (Editieren) und Adressenverzeichnis - Programme.

Argument:

Symbol in einer Formel, meistens ein Buchstabe dem eine Zahl, ein Buchstabe oder ein Name mit einer besonderen Bedeutung zugeordnet ist.

ASCII (American Standard Code for Information Interchange):

Standardcode zur Darstellung von Zahlen, Buchstaben und Symbolen. Eine ASCII-Textdatei kann, ohne Konvertierung auf einem Bildschirm angezeigt oder mit einem Drucker ausgegeben werden.

Attribut (attribute):

Dateikennzeichnung, die alternativen Charakter hat und spezielle Eigenschaften der Datei festlegt.

auf Einschalten reduzierte Anwendung (turn-key applikation):

Anwendung durch nicht rechnerorientierte Nutzer. Eine typische derartige Anwendung besteht darin, daß der Nutzer nur den Rechner einzuschalten seine eigene Platte einzulegen und das gewünschte Programm auszuwählen braucht, wobei die Auswahl aus Funktionen zu treffen ist, die auf dem Bildschirm angezeigt werden (Menü).

Aufwärtskompatibilität:

Ein für ein früher verwendetes Betriebssystem (oder Compiler usw.) erarbeitetes Programm läuft auch unter einer neuen Version des gleichen Betriebssystems.

ausführbares Programm:

Programm, das aus einer Folge von Befehlen besteht, die im vorliegenden Code im Rechner abgearbeitet werden können. Beispielsweise kann ein Rechner nicht Namen und Adressen ausführen, aber ein Programm, das diese Namen und Adressen auf Briefe schreibt.

Ausgabe:

Übertragung von Daten vom Prozessor auf das Terminal oder auf Platten.

back-up:

Kopieren einer Platte oder einer Datei zur Sicherstellung oder Erzeugung eines Platten- oder Dateiabzuges.

Begrenzer (delimiter):

Spezialzeichen, die verschiedene Teile einer Kommandozeile voneinander trennen. Im SCP trennt z.B. ein Doppelpunkt die Laufwerksbezeichnung vom Dateinamen, ein Punkt den Dateinamen vom Dateityp. Klammern in Kommandozeilen umschließen wahlweise Angaben (options), Kommas trennen die verschiedenen wahlweisen Angaben voneinander. Alle diese Sonderzeichen sind Begrenzer.

Betriebssystem BS (operating System):

Programmpaket zur Überwachung des Ablaufes anderer Programme und zur Koordinierung der verschiedenen Leistungen des Rechners. Ein BS steuert die Zusammenarbeit zwischen dem Rechner und seinen peripheren Geräten, es organisiert die Ein- und Ausgabe. Es gewährleistet eine sichere Ausführung der vom Nutzer geschriebenen Programme.

Bit:

Schalter im Speicher mit den Stellungen ein (1) oder aus (0). Bits werden in Bytes zusammengefaßt.

Block:

Für eine spezielle Verwendung reservierter Plattenspeicherbereich.

Byte:

8-bit-lange Speichereinheit.

Datei:

Zusammenstellung von Zeichen, Befehlen oder Daten, die auf einer Platte gespeichert ist. Ein Nutzer kann Plattendateien erzeugen.

Dateigruppensymbole (wildcart characters):

Im SCP gibt es zwei spezielle Sonderzeichen: ? und *. ? kann ein einzelnes Zeichen in einem Dateinamen ersetzen, * den speziellen Dateinamen, den Dateityp oder beide. Durch Eintragen der speziellen Sonderzeichen in Dateinamen entstehen mehrdeutige Dateinamen, damit können in einem Kommando mehrere Dateien angesprochen werden.

Dateiname:

Name, der einer Datei zugeordnet ist. Ein Dateiname kann den 1-8 Zeichen langen speziellen Dateinamen und den mit einem Punkt davon getrennten 0-3 Zeichen langen Dateityp umfassen.

Dateispezifikation:

Einheitliche Beschreibung einer Datei. Eine komplette SCP-Dateispezifikation umfaßt die Laufwerksbezeichnung, gefolgt von einem Doppelpunkt (d:), einen speziellen Dateinamen von 1-8 Zeichen Länge und den Dateityp mit 0-3 Zeichen Länge. b:example.tex ist ein Beispiel für eine komplette SCP-Dateispezifikation.

Dateisteuerblock (File Control Block, FCB):

Struktur, die für den Zugriff auf Plattendateien gebraucht wird. Enthält Laufwerksbezeichnung, Dateinamen, Dateityp und andere Informationen, die zur Erzeugung oder Verarbeitung einer Plattendatei benötigt werden.

Dateityp:

Erweiterung des Dateinamens. Der Dateityp kann 0-3 Zeichen lang sein und muß mit einem Punkt vom speziellen Dateinamen abgegrenzt sein. Viele Programme fordern, daß die von ihnen verarbeiteten Dateien einen vorgegebenen Dateityp haben (siehe Anlage 2).

Datendatei:

Zusammenstellung von nichtausführbaren Informationen, die durch eine Kommandodatei verarbeitet werden können.

Dienstprogramme (utility):

Hilfsprogramm zur Durchführung von Operationen, die der Nutzer benötigt, wie Datei kopieren, Datei löschen und Datei editieren. Dienstprogramme erleichtern die Arbeit der Programmierer und der Nutzer.

DIR-Attribut:

Spezielles Dateiattribut. Eine mit dem DIR-Attribut versehene Datei kann mit dem DIR-Kommando angezeigt werden. Zu beachten ist, daß nur Dateien mit aktueller Nutzernummer und vom aktuellen Laufwerk behandelt werden.

E/A (I/O):

Abkürzung für Eingabe/Ausgabe (input/output).

Editor:

Dienstprogramm zum Erzeugen und Modifizieren von Textdateien. Ein Editor kann zum Erzeugen von Schriftstücken genauso wie zur Erzeugung von Programmen für Rechner verwendet werden. Der SCP-Editor wird mit dem Kommando ED gestartet (siehe Abschnitt 5.).

Eingabe:

Übertragen von Daten in den Rechner, gewöhnlich indem sie ein Operator auf einem Terminal schreibt oder indem sie mit einem Programm von einer Platte gelesen werden.

Eingabeanforderung (prompt):

Am Bildschirm angezeigte, definierte Zeichen, die den Nutzer zur Festlegung des nächsten auszuführenden Arbeitsschrittes veranlassen. Eine System - Eingabeanforderung ist eine spezielle vom Betriebssystem angezeigte Anforderung, sie zeigt dem Nutzer, daß das Betriebssystem bereit ist, Eingaben zu empfangen. Die SCP - Eingabeanforderung besteht aus einem Buchstaben, gefolgt vom Größerzeichen (>). Der Buchstabe symbolisiert das aktuelle Plattenlaufwerk. Jedes Programm kann seine eigene spezielle Eingabeanforderung haben.

Floppy disk:

Flexible Magnetplatte (Diskette) zur Informationsspeicherung. Es gibt 5 1/4"- und 8"- Disketten.

Geräteeintragung (logged in):

Eintragen eines aktuellen Gerätes in das Betriebssystem. Ein vom Nutzer oder durch ein laufendes Programm festgelegtes Gerät wird als eingetragen bezeichnet. Es wird solange als aktuelles Gerät

verwendet, bis eine neue Eintragung erfolgt (Plattenwechsel, CTRL/C).

Hard disk:

Starre, in einem Container versiegelte Magnetplatte (Festplatte). Auf einer Festplatte können größere Informationsmengen als auf einer Diskette gespeichert werden.

Hardware:

Physische Komponenten des Rechners.

hexadezimale Notation:

Darstellung von Zahlen in einem Zahlensystem mit der Basis 16, die 16 Ziffern sind die folgenden Symbole: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E und F. Der Maschinencode wird häufig in hexadezimaler Form angegeben, weil diese Zeichen ASCII-Form haben und damit ohne Konvertierung auf dem Bildschirm oder mit Druckern ausgebbar sind (siehe Anlage 1).

Hex-Datei:

ASCII-druckbare Darstellung einer Kommandodatei (Maschinensprache).

Inhaltsverzeichnis (directory):

Bereich auf einer Platte, der Informationen über alle Dateien der Platte enthält. Die Antwort auf das DIR-Kommando ist die Anzeige aller im Inhaltsverzeichnis enthaltenen Dateinamen.

Interface:

Schnittstelle, die die Kommunikation zwischen zwei unabhängigen Systemen gestattet.

Kilobyte:

1024 Byte entsprechen 1K Byte; 1024K Byte entsprechen 1 Megabyte (MByte).

Kommando:

Vom SCP auszuführende Anweisung. Besteht aus drei Teilen: dem Kommandoschlüsselwort, der Kommandoparameterfolge und dem Endezeichen (carriage return).

Kommandodatei:

Folge von codierten, von der Maschine ausführbaren Instruktionen, die als Programmdatei auf der Platte gespeichert ist und im SCP durch Eingabe eines Kommandoschlüsselwortes auf dem Terminal nach der Eingabeanforderung gestartet wird. SCP-Kommandodateien haben den Dateityp CMD. Synonym ist Kommandoprogramm.

Kommandoparameterfolge (command tail):

Teil des Kommandos, der dem Schlüsselwort folgt. Die Kommandoparameterfolge kann eine Gerätespezifikation, einen Dateinamen und/oder Dateityp und wahlweise Parameter enthalten. Manche Kommandos benötigen keine Kommandoparameterfolge.

Kommandoschlüsselwort:

Name, der im SCP-Kommando indentifiziert, meistens der spezielle Dateiname einer Datei vom Typ CMD oder ein residentes Kommando. Dem Kommandoschlüsselwort schließen sich in der Kommandozeile eine Kommandoparameterfolge und das Endezeichen an.

Kommandosyntax:

Formale Beschreibung, die den korrekten Ablauf der Kommandoeingabe definiert. Die korrekte Struktur schließt allgemein das Kommandoschlüsselwort, die Kommandoparameterfolge und das Endezeichen ein. Eine Syntaxzeile enthält Symbole, die bei der Kommandoeingabe durch aktuelle Werte zu ersetzen sind.

Kursor:

Symbol, das auf eine Bildschirmposition zeigt. Der Kursor bestimmt die Position, auf der das nächste über die Tastatur eingegebene Zeichen erscheint.

Listengerät:

Gerät, mit dem Daten in Listenform ausgedruckt werden können.

logisches Gerät:

Ein logisches Gerät kann, muß aber nicht mit einem physischen Gerät übereinstimmen. Erläuterung an einem Beispiel: Eine Festplatte ist ein physisches Gerät. Ihr Speicherplatz kann so auf verschiedene Nutzer aufgeteilt werden, daß jedem ein eigener Bereich zugeordnet wird. Diese verschiedenen Bereiche können wie verschiedene Geräte genutzt werden. Sie werden als logische Geräte bezeichnet.

mehrdeutiger Dateiname (ambiguous filename):

Dateinamen, die im speziellen Dateinamen, im Dateityp oder in beiden die SCP-Sonderzeichen ? oder * enthalten. Werden in einem Dateinamen Zeichen durch diese Dateigruppensymbole ersetzt, so entstehen mehrdeutige Dateinamen, mit denen in einer einfachen Kommandozeile mehrere SCP-Dateien angesprochen werden können (siehe Abschnitt 2.5.).

Mikroprozessor:

Zentraleinheit (Control Processing Unit, CPU) eines Mikrorechners (besteht aus einem Chip).

Nutzernummer:

Den in einem Platteninhaltsverzeichnis aufgeführten Dateien können anwenderspezifische Nummern zugeordnet werden. Nach Angabe dieser dem jeweiligen Nutzer zugeordneten Nummer organisiert das Betriebssystem, daß der Nutzer nur mit seinen eigenen Dateien und seinem eigenen Platteninhaltsverzeichnis arbeitet, obwohl sich alle Nutzernummern auf die gleiche Platte beziehen. Im SCP können die Dateien in 16 Gruppen eingeteilt werden.

Parameter:

Größe in der Parameterfolge eines Kommandos, die dem Kommando zusätzliche Informationen liefert.

periphere Geräte:

An die CPU angeschlossene externe Geräte, wie Terminal, Drucker und Plattenlaufwerke. Sie gehören nicht zum Prozessor, werden aber in Verbindung mit ihm genutzt.

Platte, Diskette:

Medien, die zur magnetischen Speicherung von Informationen dienen. Programme und Daten werden auf einer Platte in der gleichen Weise aufgezeichnet, wie Musik auf einer Magnetbandkassette. Disketten sind auswechselbare flexible Plattenspeicher mit relativ geringer Kapazität (floppy disk). Unter dem Begriff Platte sind entweder austauschbare Disketten oder Festplatten (hard disk) zu verstehen.

Plattenlaufwerk:

Peripheres Gerät zum Schreiben auf und zum Lesen von Platten (floppy oder hard disk). Im SCP werden Geräte über zugeordnete Buchstaben angesprochen. Z.B. können vier Laufwerke einer entsprechenden Konfiguration mit A, B, C und D bezeichnet werden.

Programm:

Folge von speziell codierten Anweisungen, die bei ihrer Ausführung durch den Rechner eine vorgegebene Aufgabe erfüllen.

Puffer:

Speicherbereich, in dem während eines Informationstransfers Daten temporär gespeichert werden.

Quellendatei (source file):

ASCII-Textdatei, Eingabedatei für solche Verarbeitungsprogramme, wie Editor, Textverarbeitungsprogramm, Assembler usw.

Read-Only (RO):

Attribut, daß einer Plattendatei oder einem Plattenlaufwerk zugeordnet werden kann. Ist das Attribut einer Datei zugeordnet, kann diese gelesen, aber nicht beschrieben werden. Ist das Attribut einem Laufwerk zugeordnet, können alle auf der Platte angeordneten Dateien gelesen werden. Das Eintragen einer neuen Datei, das Löschen oder Ausstauschen einer Datei, das Ändern des Namens einer Datei oder das Schreiben auf die Platte werden verhindert. Eine Datei oder ein Laufwerk kann mit dem STAT-Kommando in dem RO-Zustand gesetzt werden. Das Gegenteil von RO ist RW (Read-Write). Eine Datei oder ein Laufwerk haben entweder den Zustand RO oder RW. Der Standardzustand der Geräte und Dateien ist RW, tritt aber beim Rücksetzen oder Wechseln der Platte ein Fehler auf, so wird automatisch das Gerät auf RO gesetzt.

Read-Write (RW):

Attribut, daß einer Plattendatei oder einem Plattenlaufwerk zugeordnet werden kann. Ist das Attribut RW einer Plattendatei zugeordnet, so kann sie gelesen und geschrieben werden. Ist RW einem Plattenlaufwerk zugeordnet, so können alle Dateien der in diesem Laufwerk befindlichen Platte gelesen oder geschrieben werden.

residentes Kommando (built-in command):

Kommando, das sich permanent im Speicher befindet. Ist sofort verfügbar, weil es nicht erst von der Platte geholt werden muß.

Satz (record):

Zusammenstellung von Daten. Eine Datei besteht aus einem oder mehreren Sätzen, die auf der Platte gespeichert sind. SCP-Sätze sind 128 Bytes lang.

Sektor:

Abschnitt einer Plattenspur. Jede Spur hat eine festgelegte Anzahl von Sektoren.

Software:

Programmausrüstung des Rechners, im Gegensatz zur Hardware, die die physischen Komponenten des Rechners bezeichnet.

spezieller Dateiname (primary filename):

Der spezielle Dateiname identifiziert eine Datei und dient dem Nutzer als Hinweis auf ihren Inhalt. Er ist 1-8 Zeichen lang und kann aus Buchstaben, Ziffern und Sonderzeichen zusammengesetzt sein. Der spezielle Dateiname steht zwischen der wahlweisen Gerätespezifikation und der wahlweisen Angabe eines Datentyps.

Spur (track):

Konzentrischer Ring auf der Platte. Die üblichen 8" - Disketten haben 77 Spuren.

Steuerzeichen:

Nichtdruckbare Zeichenfolge, die ein einfaches SCP-Kommando beinhaltet. Einige Steuerzeichen veranlassen zeilenbezogene Editierfunktionen. Die Eingabe eines Steuerzeichens umfaßt das Drücken und Festhalten der CONTROL-Taste des Teminals und das Anschlagen einer spezifischen Buchstabentaste (siehe Anlage 3).

Syntax:

Formale Beschreibung des Kommandoaufbaus.

Systemattribut:

Das Dateiattribut SYS kann einer Datei mit dem STAT-Kommando zugeordnet werden. Dateien mit dem Attribut SYS werden nicht durch das DIR-Kommando angezeigt, sondern durch das Kommando DIRS (siehe Abschnitt 4.). Eine Datei mit der Nutzernummer 0 und dem Attribut SYS kann von allen Nutzerbereichen des gleichen Laufwerkes aus gelesen und ausgeführt werden. Damit können allgemein benötigte Programme für alle Nutzer mit verschiedenen Nummern zur Verfügung gestellt werden.

System - Eingabeanforderung (system prompt):

Auf dem Bildschirm angezeigtes Symbol, das die Eingabebereitschaft des Betriebssystems signalisiert (vgl. Eingabeanforderung).

Terminal:

Hauptein/ausgabegerät. Das Terminal besteht aus einem Listenausgabegerät, einem Bildschirm und einer Tastatur. Über ein Terminal erfolgt die Kommunikation der Nutzer mit dem Betriebssystem oder einem Anwenderprogramm.

Verketten (concatenate):

Term, der eine PIP-Operation beschreibt, die zwei oder mehr separate Dateien zu einer neuen Datei in einer spezifizierten Reihenfolge kopiert.

Wahlweise Angaben (options):

Mögliche zusätzliche Angaben in der Parameterfolge eines Kommandos. Damit werden zusätzliche spezielle Bedingungen für die Kommandoausführung festgelegt.

## Anlage 6 Erweiterung der HELP.HLP-Datei

HELP arbeitet mit zwei in Beziehung stehenden Dateien, der HELP.CMD-Datei und der HELP.HLP-Datei.
Die Kommandodatei (HELP.CMD) steuert den Zugriff auf die Textdatei (HELP.HLP) und gibt den Text über Bildschirm aus. Der Textdatei (HELP.HLP) kann Text hinzugefügt werden, sie ist aber nicht editierbar.
Unter Verwendung der Kommandodatei HELP.CMD muß erst die Textdatei HELP.HLP in die Textdatei HELP.DAT konvertiert werden, bevor diese dann editiert oder eigener Text eingefügt werden kann. Für die Konvertierung von HELP.HLP zu HELP.DAT und wieder zurück können folgende Formen des HELP-Kommandos genutzt werden:

```
HELP[E]        (HELP.HLP ---> HELP.DAT)
HELP[C]        (HELP.DAT ---> HELP.HLP)
```

HELP[E] greift auf die Datei HELP.HLP auf dem Standardlaufwerk zu, löscht den Kopfsatz und erstellt die Datei HELP.DAT auf demselben Laufwerk. Danach kann die Datei HELP.DAT editiert oder eigener Text hinzugefügt werden.
HELP[C] greift auf die editierte Datei HELP.DAT zu, generiert den ersten Datensatz neu, und erstellt eine neue Datei HELP.HLP.
Von HELP.CMD kann der neue Text der HELP.HLP-Datei ausgegeben werden.
Das Hinzufügen von Begriffen (Topics) und Unterbegiffen (Subtopics) in die Datei HELP.HLP erfolgt in einem spezifischen Format. Das allgemeine Format einer Begriffsüberschrift in der Datei HELP.DAT hat folgende Form.

```
///nTOPICNAME(CR)
```

Die drei Schrägstriche sind der Begriffsbegrenzer und müssen in Spalte 1 beginnen, n ist eine Ziffer von 1 bis 9 und steht für die Ebene des Begriffes. Ein Hauptbegriff hat immer die Nummer 1. Der erste Unterbegriff hat die Nummer 2. Der nächste untergeordnete Unterbegriff hat dann die Nummer 3 usw. bis zu einem Maximum von 9 Ebenen.
TOPICNAME ist der Name des einzufügenden Begriffes und kann bis zu 12 Zeichen lang sein. Die Zeile wird mit Wagenrücklauf (CR) beendet.

Folgende Hinweise beim Editieren und Einfügen von Text in die Datei HELP.DAT sind zu beachten.

- Begriffe sollten alphabetisch geordnet sein.
- Unterbegriffe sollten innerhalb ihres entsprechenden Hauptbegriffes alphabetisch geordnet sein.
- Ebenen müssen mit Ziffern zwischen 1 und 9 versehen werden.

Ein Beispiel für Hauptbegriff und Unterbegriff in der Datei HELP.HLP ist:

```
///1NEW UTILITY(CR)
///2COMMANDS(CR)
///3EXAMPLES(CR)
```

Das Beispiel zeigt das Format eines Hauptbegriffes (main topic) mit der Ebenennummer 1. Die zweite Zeile zeigt, wie ein Unterbegriff zu diesem Hauptbegriff zu numerieren ist. Die dritte

Zeile zeigt, wie die nächste Ebene eines Unterbegriffes zu dem Hauptbegriff numeriert werden muß.
Beim Abarbeiten des HELP-Kommandos müssen genügend Zeichen nach der Kommandoanforderung eingegeben werden, damit HELP eindeutig den TOPICNAME identifizieren kann, also beispielsweise

```
HELP>ED
```

Wenn die Informationen zu einem Unterbegriff ausgegeben werden sollen, muß der Hauptbegriff und der Unterbegriff angegeben werden, sonst kann HELP nicht entscheiden, welcher Hauptbegriff gemeint ist, also beispielsweise

```
HELP>ED COMMANDS
```

Dieses Kommando gibt die Informationen über die internen Kommandos des Editors aus.

## Sachwortverzeichnis

Attribut 17

Dateigruppensymbol 16, 26
Dateiname 25
Dateispezifikation 14, 25
Dateisteuerblock 60
Dateityp 25
Dateiverzeichnis 13
Dienstprogramm 21

Eingabeanforderung 8
Eingabemodus 40

Gerätename, logisch 51, 64
Gerätename, physisch 64

Kommando, resident 8
Kommando, transient 8
Kommandomodus 37
Kommandoschlüsselwort 8

Laufwerkskennung 25

Monitorprogramm 7

Nutzernummer 17, 47

Paragraphwert 42

SCP-Kommando 8
Speicherplatte 7
Stapelverarbeitung 65
Steuerzeichen 9
Suchbegriff 43

Texteditor 37
Trennzeichen 14

Zeichenzeiger 37, 75